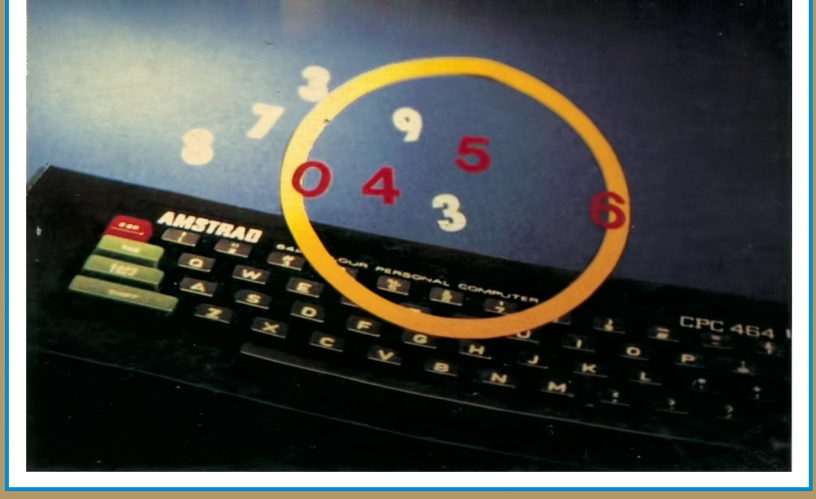


P. BEAUFILS / M. LAMARCHE / Y. MUGGIANU

PROGRAMMES DE MATHEMATIQUES SUR AMSTRAD

EYROLLES

PROGRAMMES DE MATHEMATIQUES SUR AMSTRAD

CHEZ LE MEME EDITEUR

Des mêmes auteurs :

— *Programmes de physique sur Amstrad.*

Autres ouvrages :

C. DELANNOY — *Faites vos jeux avec Amstrad.*

— *Je débute en BASIC Amstrad.*

P. BIHAN — *Programmation sur AMSTRAD PCW 8256 et 8512 Basic et fichiers.*

M. ROUSSELET — *Calcul numérique sur Amstrad.*

O. LEPAPE — *L'assembleur facile du Z 80.*

Ph. DAX — *CP/M et sa famille. Guide d'utilisation.*

A. REVERCHON, M. DUCAMP

— *Mathématiques sur micro-ordinateur.*
1 - Analyse.
2 - Algèbre.

F. MANCHON, J.-M. NASR

— *Physique sur micro-ordinateur.*

J.-P. DELAHAYE

— *Dessins géométriques et artistiques avec votre micro-ordinateur.*

— *Nouveaux dessins géométriques et artistiques avec votre micro-ordinateur.*

M. JAMES — *Introduction à l'intelligence artificielle sur micro-ordinateur.*

PROGRAMMES DE MATHEMATIQUES SUR AMSTRAD

par

P. BEAUFILS / M. LAMARCHE / Y. MUGGIANU

EYROLLES

61, boulevard Saint-Germain — 75005 PARIS
1986

Si vous désirez être tenu au courant de nos publications, il vous suffit d'adresser votre carte de visite au :

Service « Presse », Editions EYROLLES,
61, Boulevard Saint-Germain
75240 PARIS CEDEX 05,

en précisant les domaines qui vous intéressent. Vous recevrez régulièrement un avis de parution des nouveautés en vente chez votre libraire habituel.

« La loi du 11 mars 1957 n'autorisant, aux termes des alinéas 2 et 3 de l'article 41, d'une part, que les « copies ou reproductions strictement réservées à l'usage privé du copiste et non destinées à une utilisation collective » et, d'autre part, que les analyses et les courtes citations dans un but d'exemple et d'illustration, « toute représentation ou reproduction intégrale, ou partielle, faite sans le consentement de l'auteur ou de ses ayants droit ou ayants cause, est illicite » (alinéa 1er de l'article 40). »

« Cette représentation ou reproduction, par quelque procédé que ce soit, constituerait une contrefaçon sanctionnée par les articles 425 et suivants du Code pénal. ».

© Editions EYROLLES, 1986

Préface

Nous sommes physiciens et pourtant le titre de cet ouvrage parle de Mathématiques, ceci mérite une explication.

Ce livre est initialement écrit par des physiciens pour des physiciens :

– pour éviter qu'ils soient bloqués dans la résolution de leurs problèmes par des difficultés mathématiques et par là leur suggérer de s'attaquer à des problèmes encore plus complexes ;

– pour leur fournir des méthodes de calcul numérique ou d'approximations rapides et efficaces;

– pour leur faire découvrir par l'exemple des domaines des mathématiques qu'ils connaissent peut-être mal.

Certes les résultats fournis par l'ordinateur ne sont qu'approchés, mais le physicien préfère souvent une solution approximative plutôt qu'une absence de solution exacte. Il faut dire que les approximations faites dans les calculs sont parfois bien inférieures à celles consistant à négliger certains phénomènes physiques secondaires.

Mais nous avouons avoir poursuivi également un autre but : intéresser nos collègues mathématiciens aux possibilités de l'ordinateur. Nous les soupçonnons en effet de penser que l'usage de l'ordinateur «tue» les mathématiques. Nous n'en croyons rien : la conception des programmes et la critique nécessaire des résultats qu'ils fournissent ouvrent aux mathématiciens un grand champ de travail.

Certes, nous concevons qu'ils bondissent lorsque l'AMSTRAD appelle PI un nombre rationnel. Remarquez toutefois qu'il n'accepte pas PI ou $\sqrt{2}$ à une demande d'introduction de données ou qu'il ne fournit pas PI pour résultat. Dans la mesure où l'on aboutira à une solution numérique approchée du problème, cet inconvénient n'est pas grave, mis à part la question de langage.

Mais par d'autres côtés, l'utilisation de l'ordinateur fait progresser dans la précision de l'écriture. Ainsi $\sin^2 x + 3$ doit disparaître au profit par exemple de $(\sin (x)) \uparrow 2 + 3$, ce qui est un bien. De même, l'usage du symbole * pour la multiplication évite des erreurs et permet l'emploi de variables désignées par deux lettres telles que «ab», ce qui est bien pratique.

Quel peut être l'intérêt pour le mathématicien d'une solution numérique approchée ? Elle lui permet de déceler une solution sans intérêt ou par son invraisemblance de trahir une erreur de calcul. La solution numérique, le graphe d'une fonction par exemple, sont à l'expression littérale ce que la photographie d'identité est au visage : ce n'est pas le visage, mais c'est très utile pour reconnaître un visage.

Bien qu'il soit spécialisé dans l'application numérique, l'ordinateur privilégie la notion de fonction, ce qui en fait en ce domaine un instrument pédagogique puissant. En voici un exemple : soit à dessiner sur l'écran des vaguelettes, ou un trèfle à quatre feuilles, ou un serpent ; si vous définissez ces figures point par point, il vous faudra introduire des centaines de nombres. Si vous les concevez comme graphes de fonction, il suffit d'écrire leur équation et de préciser le domaine de la variable. Toute personne ayant fait cette démarche sera convaincue de l'intérêt de la notion de fonction.

Venons-en enfin à la critique des résultats. Si vous tracez le graphe de $y = \sin (4.x)$ en prenant pour x des valeurs entières ou si vous intégrez cette fonction avec $dx = 1$, vous obtiendrez un résultat fantaisiste. Il est très difficile de réaliser un programme *incollable* et si le bon sens permet de rejeter la solution improbable, il faut parfois un sens critique et mathématique très aigu pour rejeter une solution apparemment convenable, mais en réalité fausse.

LES AUTEURS

Sommaire

Graphes de fonctions Y (X) et R (A) 1
Graphisme mathématique 9
Primitives et équations différentielles 28
Equations différentielles du second ordre, complètes, linéaires et à coefficients constants 35
Méthode de Runge-Kutta. Résolution d'équations différentielles 40
Spirale de Cornu 44
Le voyage Terre-Lune selon Jules Verne 47
Calcul d'une intégrale définie. Méthode de Simpson 53
Calcul d'une intégrale définie. Méthode des Trapèzes 55
Recherche des zéros d'un polynôme. Méthode de Bairstow 57
Statistiques et calcul de probabilité. Loi de Gauss 65
Système d'équations linéaires 73
Changement de base 81
Logique 83
Séries de Fourier 89
Transformée de Fourier rapide 102
Nombres aléatoires 119
Valeur moyenne. Valeur quadratique moyenne 137
Représentation d'une surface z (x, y) 144
Représentation d'une surface en coordonnées cylindriques 152
Représentation « Fil de Fer » de polyèdres 162
Fractal 169
Méthode des moindres carrés. Applications aux équations non linéaires 175

Ces programmes ont été écrits pour les ordinateurs AMSTRAD. Les commentaires qui les accompagnent permettent de les adapter facilement à d'autres ordinateurs travaillant en BASIC.

Les illustrations ont été réalisées avec l'imprimante AMSTRAD CMP 2000.

La compréhension des commentaires requiert le niveau mathématique d'une classe terminale scientifique.

Graphes de fonctions
Y (X) ET R (A)

Il ne s'agit pas ici de découvrir une fonction quelconque sur l'intervalle $-\infty$, $+\infty$, mais d'obtenir une représentation graphique d'une fonction connue sur un intervalle bien déterminé. En particulier, la fonction doit être définie sur cet intervalle et ne pas présenter d'accident sur un domaine égal à 1/600 du domaine d'étude. Nous vous présentons donc un programme fait pour une fonction particulière, mais les commentaires qui suivent vous permettront de l'adapter facilement à toute autre fonction de votre choix.

Tracé de Y_1 (X)

Nous avons d'abord choisi de représenter la fonction numéro :

$y_1 = 8 \,.\, \exp(-x) \,.\, \sin(8 \,.\, x)$ (ligne 40)

x variant de $xa = 1$ à $xb = 3$ (ligne 60)

L'étendue ix du domaine d'étude fixe l'échelle en abscisse ex (ligne 120). L'intervalle entre deux valeurs de x considérées est dx (ligne 1030) $= ix/600$. Le nombre 600 est fixé par le nombre de pixels sur une largeur d'écran compte tenu des marges. Il est inutile de calculer plus de valeurs car les points correspondants ne se distingueraient pas sur l'écran.

Abordons maintenant le problème de l'échelle en ordonnées :

Le sinus est compris entre -1 et $+1$, l'exponentielle décroissante prendra sa valeur maximale pour $x = 1$, elle est inférieure à 0,37, multiplié par 8, cela fait moins de 3. y est donc compris entre -3 et $+3$ dans le domaine considéré. On peut bien sûr demander à l'ordinateur de déterminer ce domaine de valeur de y en calculant y pour les 600 valeurs de x et en recherchant les valeurs maximale et minimale. Ce procédé est

utilisé dans les programmes «valeur moyenne» et «primitives», nous ne l'aborderons donc pas ici, mais remarquez que notre petit raisonnement mathématique a évité un calcul laborieux à l'ordinateur.

Connaissant les valeurs extrémales *yp* et *ym* (ligne 80), on en déduit l'ampleur *iy* de l'intervalle de valeur de *y* et compte tenu de la hauteur de l'écran (360 en ligne 150), l'échelle *ey* . [[*by* est le décalage qui va nous permettre de placer l'axe des abscisses. Si la fonction étudiée garde le même signe, positif, par exemple, on pourra choisir *ym* = 0 afin d'observer l'axe des abscisses sur l'écran.

Tout ceci étant fait, la fonction est calculée pour chaque valeur de *x* (ligne 1050). Les coordonnées du point représentatif sont calculées compte tenu des marches, de l'échelle et du décalage en ligne 1080 et 1090 (fig. 1).

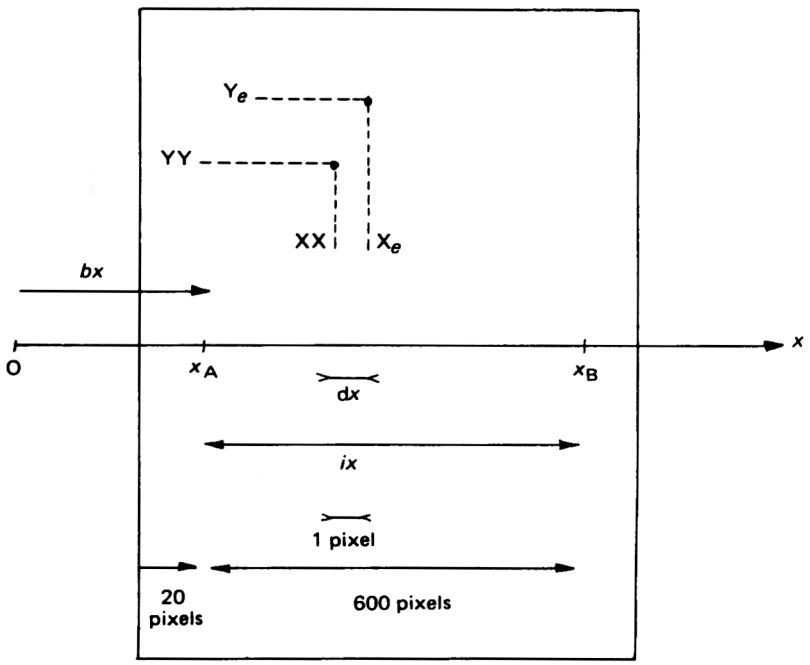


Fig. 1.

Si vous supprimez les lignes 1302 à 1306, vous obtenez sur l'écran une distribution de points suggérant la courbe. En ajoutant ces lignes, vous joignez ces points par des droites (théoriquement, en fait ce sont des courbes en escalier). A vous de choisir entre une suggestion de courbe assez exacte et une courbe imposée un peu moins exacte (fig. 2).

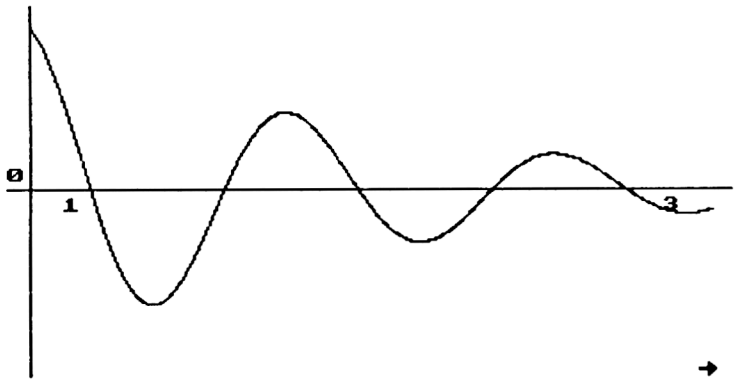


Fig. 2.

Tracé de Y_2 (X)

Lorsque la flèche apparaît, vous pouvez appuyer sur la touche correspondante pour observer le graphe de la deuxième fonction obtenu de la même manière que le précédent (fig. 3) :

$$y = \exp(x/8) \,.\, \sin(2 \,.\, x)$$

x variant de -3 à $+3$

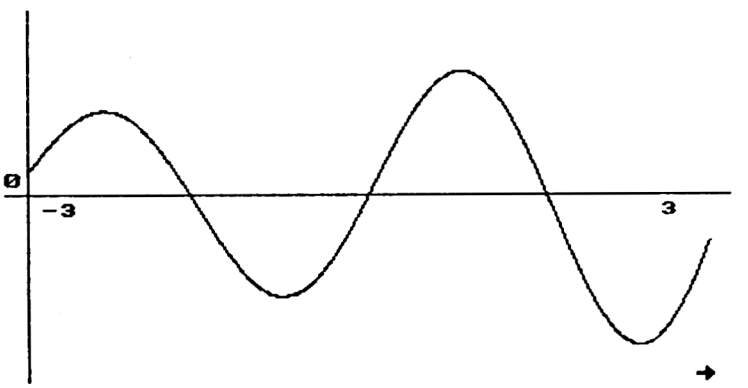


Fig. 3.

Tracé de Y_2 (Y_1 (X))

L'étape suivante du programme montre (fig. 4) le graphe de la fonction composée à partir des deux précédentes, x variant de 1 à 3. Lorsque la variable prend la valeur x, la première fonction en donne y_1 qui est la valeur de la variable pour la seconde fonction (ligne 260).

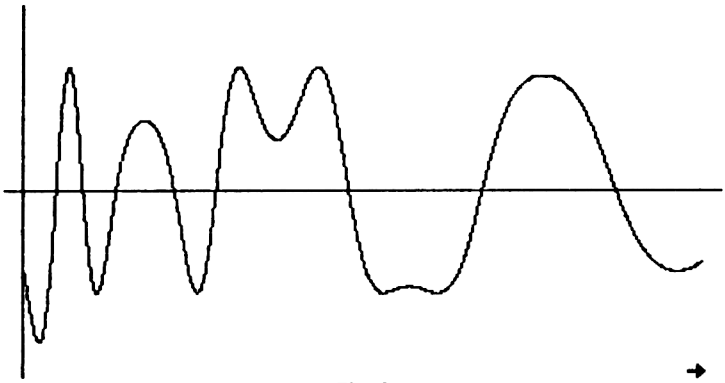


Fig. 4.

Tracé d'une courbe définie paramétriquement

Un point de cette courbe a pour abscisse y_1 (x) et pour ordonnée y_2 (x), nous avons représenté figure 5 la portion de courbe correspondant à une variation de x de 1 à 3.

Structure du programme

Ligne(s)	
40	définition des fonctions
60	préparation des graphes
170	envoi à 1000 pour le tracé de la courbe
1000	tracé de y_1 ou de y_2
200	tracé de la fonction composée
340	tracé de la courbe paramétrique

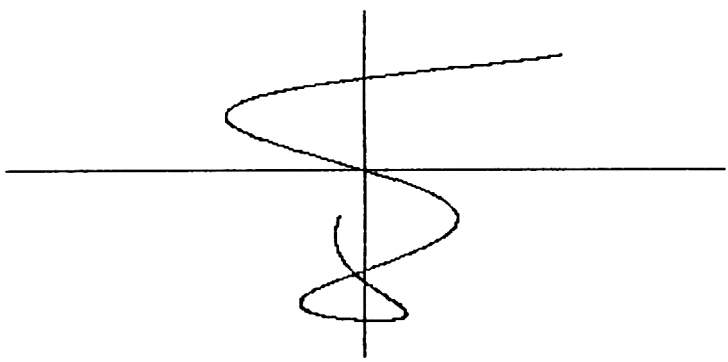


Fig. 5.

Liste des variables

x variable
n numéro de la fonction
y valeur de la fonction
xa première valeur de *x*
xb dernière valeur de *x*
ym, yp valeurs minimale et maximale de *y*
ex, ey échelles en *x* et en *y*
ix, iy largeurs des domaines de valeurs de *x* et *y*
bx, by décalages en abscisses et en ordonnées
dx intervalle entre deux valeurs de *x* consécutives
xe, ye coordonnées du point sur l'écran
xx, yy coordonnées du point précédemment marqué
v valeur de la variable dans le cas des fonctions composées
dv écart entre deux valeurs de *v*

Premier Programme

```
10 REM"graphes de fonctions"
20 REM
30 CLS:GOTO 60
40 y(1)=8*EXP(-x)*SIN(8*x)
45 RETURN
50 y(2)=EXP(x/8)*SIN(2*x)
55 RETURN
60 xa(1)=1:xb(1)=3
70 xa(2)=-3:xb(2)=3
80 ym(1)=-3:yp(1)=3
90 ym(2)=-1.5:yp(2)=1.5
100 FOR n=1 TO 2
110 ix(n)=xb(n)-xa(n)
120 ex(n)=600/ix(n)
130 bx(n)=-xa(n)*ex(n)
140 iy(n)=yp(n)-ym(n)
150 ey(n)=360/iy(n)
160 by(n)=-ym(n)*ey(n)
170 GOSUB 1000
180 NEXT n
190 REM
200 MOVE 0,20+by(2):DRAWR 640,0
210 MOVE 20,0:DRAWR 0,400
220 dv=ix(1)/600
230 FOR v=xa(1) TO xb(1) STEP dv
240 x=v:xe=20+bx(1)+ex(1)*x
250 GOSUB 40
260 x=y(1)
270 GOSUB 50
280 ye=20+by(2)+ey(2)*y(2)
290 PLOT xe,ye
292 IF v=xa(1) THEN GOTO 296
294 DRAWR xx-xe,yy-ye
```

```
296 xx=xe:yy=ye
300 NEXT v
310 LOCATE 39,25:PRINT CHR$(243)
320 q=INKEY(1):IF q<>0 THEN 320
330 REM
340 CLS
350 MOVE 320,0:DRAWR 0,400
360 MOVE 0,200:DRAWR 640,0
380 FOR x=xa(1) TO xb(1) STEP ix(1)/600
390 GOSUB 40
400 GOSUB 50
410 xe=320+ey(1)*y(1)
420 ye=200+ey(2)*y(2)
430 PLOT xe,ye
432 IF x=xa(1) THEN GOTO 436
434 DRAWR xx-xe,yy-ye
436 xx=xe:yy=ye
440 NEXT x
970 END
980 REM
990 REM
1000 MOVE 0,20+by(n):DRAWR 640,0
1010 MOVE 20,0:DRAWR 0,400
1020 LOCATE 3,24-(by(n)/16)+1:PRINT STR$(xa(n))
1030 LOCATE 36,24-(by(N)/16)+1:PRINT STR$(xb(n))
1040 LOCATE 1,24-(by(n)/16)-1:PRINT"0"
1050 dx(n)=ix(n)/600
1060 FOR x=xa(n) TO xb(n) STEP dx(n)
1070 GOSUB 40:GOSUB 50
1080 xe=20+bx(n)+ex(n)*x
1090 ye=20+by(n)+ey(n)*y(n)
1300 PLOT xe,ye
1302 IF x=xa(n) THEN GOTO 1306
1304 DRAWR xx-xe,yy-ye
1306 xx=xe:yy=ye
1310 NEXT x
1320 LOCATE 39,25:PRINT CHR$(243)
1330 q=INKEY(1):IF q<>0 THEN 1330
1340 CLS
1350 RETURN
```

Coordonnées polaires

La courbe est tracée après avoir choisi l'échelle ($e = 50$ en ligne 80) et le domaine de variation de l'angle (ligne 60 à 70). Le point est placé en lignes 130 à 150. Il est relié au précédent en lignes 160 à 180 (fig. 6).

Vous prendrez garde que certaines imprimantes, dont celle-ci, déforment l'image. Pour obtenir sur le papier une image non déformée, il faut multiplier *ye* par un coefficient adéquat. Vous mesurerez le coefficient de distorsion en faisant recopier par l'imprimante un carré tracé sur l'écran.

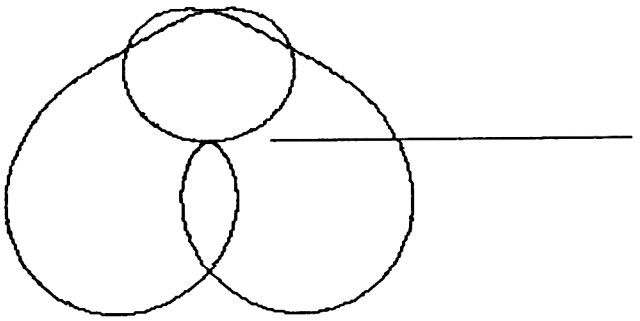


Fig. 6.

Deuxième Programme

```
10 REM"COURBE EN POLAIRE"
20 REM
30 CLS:RAD:GOTO 60
40 R=4*SIN(a/2)+SIN(2*a)
50 RETURN
60 a1=-2*PI
70 a2=2*PI
80 e=50
90 ia=a2-a1:da=ia/500
100 MOVE 320,200:DRAWR 320,0
110 FOR a=a1 TO a2 STEP da
120 GOSUB 40
130 xe=320+e*r*COS(a)
140 ye=200+e*r*SIN(a)
150 PLOT xe,ye
160 IF a=a1 THEN GOTO 180
170 DRAWR xx-xe,yy-ye
180 xx=xe:yy=ye
190 NEXT a
```

Dans la courbe de la figure 6, on peut reconnaître un demi cœur correspondant à une variation de *a* de — pi à pi. Tournons la courbe de pi/2, modifions le domaine d'étude en conséquence, prenons la valeur absolue de l'angle pour obtenir la courbe symétrique et nous avons le cœur. En fait nous avons été amenés à modifier quelque peu l'équation pour creuser le cœur dans sa partie inférieure (fig. 7). Les corrections à apporter sont donc les suivantes :

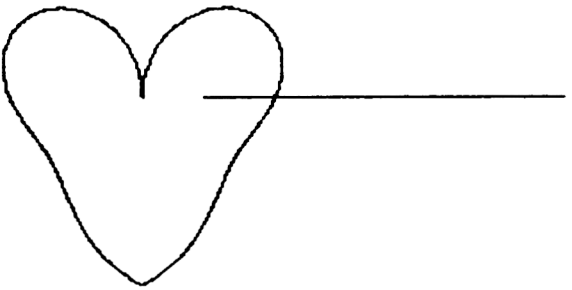


Fig. 7.

Troisième Programme

```
10 REM"COURBE EN POLAIRE"
20 REM
30 CLS:RAD:GOTO 60
40 aa=ABS(a-PI/2)
41 R=4*SIN(aa/2)+SIN(2*aa)*(1+0.3*SIN(aa))
50 RETURN
60 a1=-PI/2
70 a2=PI*3/2
80 e=50
90 ia=a2-a1:da=ia/500
100 MOVE 320,200:DRAWR 320,0
110 FOR a=a1 TO a2 STEP da
120 GOSUB 40
130 xe=320+e*r*COS(a)
140 ye=200+e*r*SIN(a)
150 PLOT xe,ye
160 IF a=a1 THEN GOTO 180
170 DRAWR xx-xe,yy-ye
180 xx=xe:yy=ye
190 NEXT a
```

Graphisme mathématique

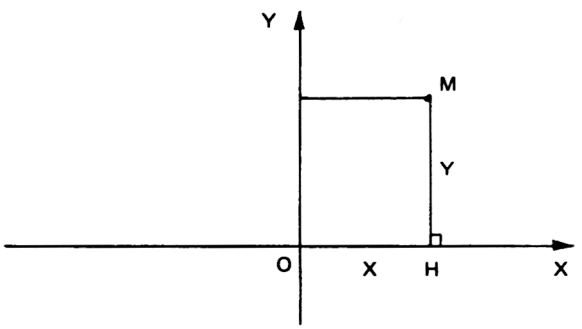
LES DIFFÉRENTS SYSTÈMES DE COORDONNÉES

Il y a trois principaux systèmes de coordonnées que l'on emploie pour représenter graphiquement les fonctions et les courbes.

1. Les coordonnées cartésiennes :

Un point M du plan est déterminé par son abscisse X et par son ordonnée Y.

On posera $\overline{OH} = X$ et $\overline{HM} = Y$.



L'équation est alors donnée soit sous forme explicite $Y = f(X)$, soit sous forme implicite $f(X, Y) = 0$.

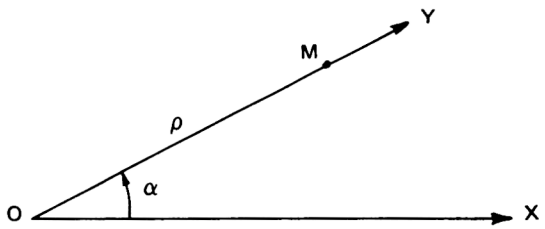
2. Les coordonnées polaires :

Les coordonnées polaires d'un point M sont définies à partir d'un axe OX, appelé axe polaire origine, contenant le point O appelé pôle.

Si OY est un axe passant par M, on définit l'angle des axes : (OX, OY) = α et la mesure algébrique $\overline{OM} = \rho$.

α et ρ constituent les coordonnées polaires du point M.

La relation $\rho = f(\alpha)$ est l'équation polaire d'une courbe.



3. Les coordonnées paramétriques :

La position d'un point M est donnée par son abscisse X et son ordonnée Y, mais chacune d'elles étant exprimée en fonction d'un paramètre, lequel est en général un angle α, ou le temps *t*.

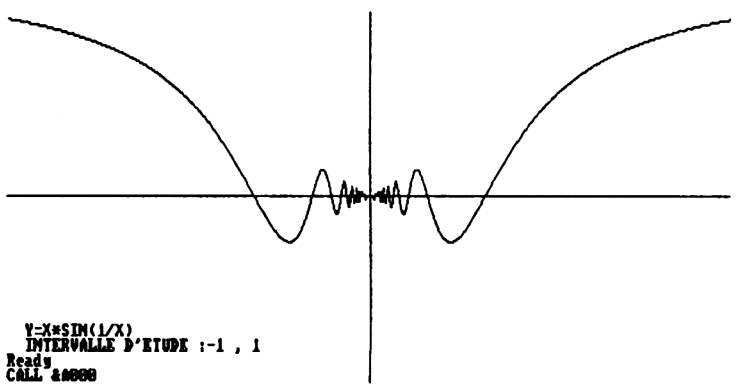
Programme 1

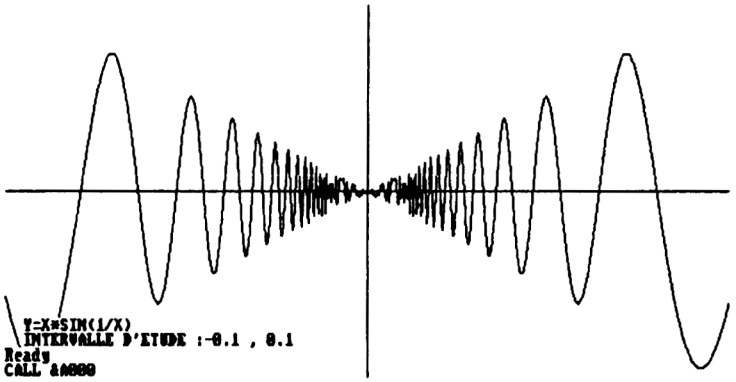
Ce programme va vous permettre de dessiner la courbe de toutes les fonctions en coordonnées cartésiennes.

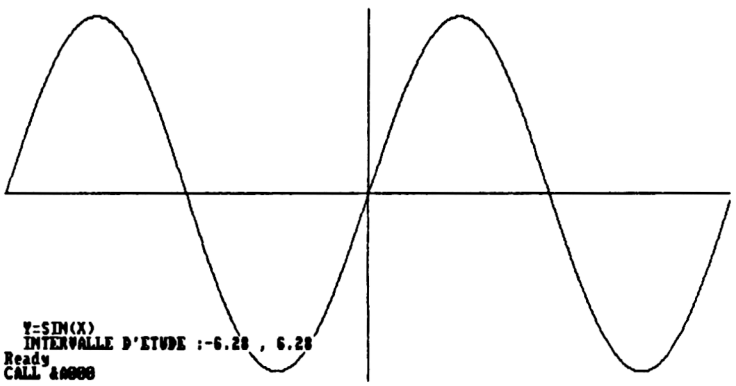
Déroulement du programme

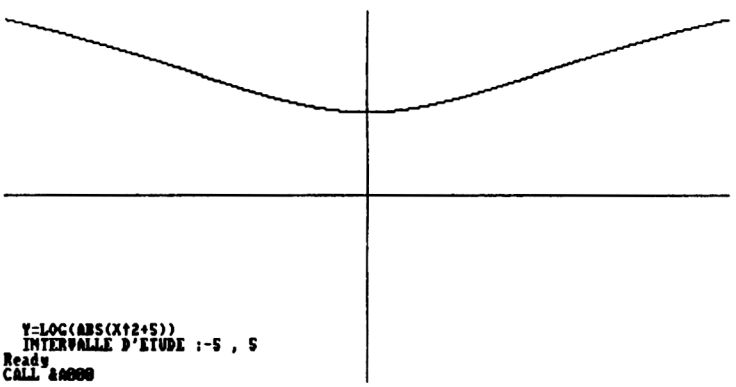
Ligne (s)	
40	l'ordinateur demande une chaîne A$ qui est la fonction à étudier
50	l'ordinateur vous demande de lui fournir un nombre N. Il tracera la courbe pour les valeurs de la variable comprises entre — N et + N (voir ligne 100)
80 à 200	l'ordinateur ouvre un tableau de 640 valeurs. La fonction est donnée en ligne 120. Une échelle automatique est calculée par l'ordinateur (ligne 130). En ligne 110, on rentre les conditions particulières nécessaires à l'étude de la fonciton
300 à 330	on trace les axes
340 à 430	l'ordinateur trace la courbe en continu (et non point par point)

```
10 REM FONCTION Y=F(X)
20 REM ***************
30 MODE 2
40 INPUT "ETUDE DE LA FONCTION Y=";A$
50 INPUT "INTERVALLE D'ETUDE N=";N
60 CLS
70 LOCATE 15,12:PRINT "CALCUL EN COURS"
80 DIM A(640):S=0
90 FOR F=0 TO 640
95 LOCATE 3,15:PRINT F
100 X=(F-320)*N/320
110 IF X=0 THEN GOTO 200
120 A(F)=X*SIN(1/X)
130 IF ABS(A(F))>S THEN S=ABS(A(F))
200 NEXT F
210 CLS
300 REM TRACE DE LA FONCTION
310 REM ********************
320 PLOT 320,0:DRAWR 0,400
330 PLOT 0,200:DRAWR 640,0
340 T=0
350 FOR F=0 TO 640
360 IF ABS(A(F))/S=1 THEN T=0:GOTO 400
370 Y=A(F)/S*190
380 IF T=0 THEN PLOT F,Y+200:T=1:GOTO 400
390 DRAWR 1,Y-AY
400 AY=CINT(A(F)/S*190)
410 NEXT F
420 LOCATE 3,22:PRINT "Y=";A$
430 LOCATE 3,23:PRINT "INTERVALLE D'ETUDE :";-N;",
";N
```




Y=X*SIN(1/X)
INTERVALLE D'ETUDE :-0.1 , 0.1
Ready
CALL &A000


Y=SIN(X)
INTERVALLE D'ETUDE :-6.28 , 6.28
Ready
CALL &A000

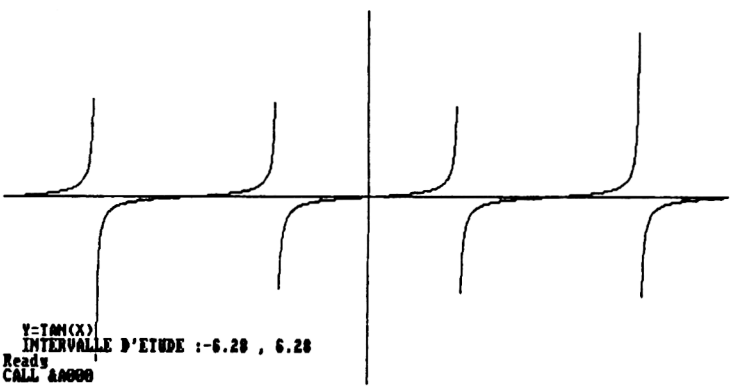

Y=LOG(ABS(X↑2+5))
INTERVALLE D'ETUDE :-5 , 5
Ready
CALL &A000

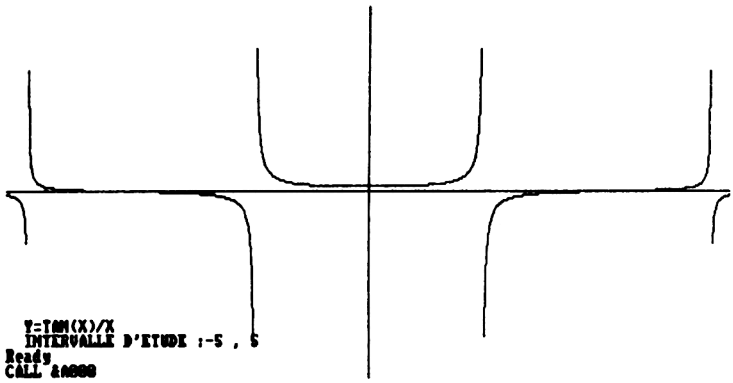
Programme 2

Il présente l'étude d'une fonction Y = TAN (X) qui comprend donc des branches asymptotiques.

Observez particulièrement les lignes 125 et 360.

```
10 REM FONCTION Y=F(X)
20 REM ***************
30 MODE 2
40 INPUT "ETUDE DE LA FONCTION Y=";A$
50 INPUT "INTERVALLE D'ETUDE N=";N
60 CLS
70 LOCATE 15,12:PRINT "CALCUL EN COURS"
80 DIM A(640):S=0
90 FOR F=0 TO 640
95 LOCATE 3,15:PRINT F
100 X=(F-320)*N/320
120 A(F)=TAN(X)
125 IF ABS(A(F))>50 THEN A(F)=0:GOTO 200
130 IF ABS(A(F))>S THEN S=ABS(A(F))
200 NEXT F
210 CLS
300 REM TRACE DE LA FONCTION
310 REM ********************
320 PLOT 320,0:DRAWR 0,400
330 PLOT 0,200:DRAWR 640,0
340 T=0
350 FOR F=0 TO 640
360 IF A(F)=0 OR ABS(A(F))/S=1 THEN T=0:GOTO 400
370 Y=A(F)/S*190
380 IF T=0 THEN PLOT F,Y+200:T=1:GOTO 400
390 DRAWR 1,Y-AY
400 AY=CINT(A(F)/S*190)
410 NEXT F
420 LOCATE 3,22:PRINT "Y=";A$
430 LOCATE 3,23:PRINT "INTERVALLE D'ETUDE :";-N;",
";N
```





Programme 3

Ce programme vous permettra de calculer l'intersection de la courbe avec l'axe des X par la méthode de dichotomie.

MÉTHODE DE DICHOTOMIE

Considérons l'équation sin X = 0.2 X — 0.5, la variable X étant exprimée en radians. On ne sait pas la résoudre par les méthodes usuelles algébriques. Habituellement on trace Y = sin X, Z = 0.2 X — 0.5 et on a la solution graphique au point d'intersection.

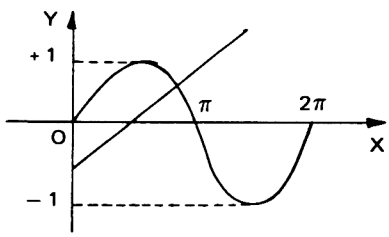


Fig. 1.

Une autre méthode consiste à essayer des valeurs et à tenter de déterminer pour quelle valeur de X, sinX s'approche de 0.2 X — 0.5, ou encore pour quelle valeur de X l'expression :
«sin X — 0.2 X + 0.5» est nulle.

Pour certaines valeurs de X, l'expression précédente (appelons-la *f* (X)) est positive, pour d'autres elle est négative. On a ainsi deux bornes de X, soit A pour l'inférieure et B pour la supérieure entre lesquelles existe la racine. L'astuce consiste à resserrer la fourchette (A, B).

Remarque importante :

La racine se trouve dans l'intervalle A, B si f (A) et f (B) sont de signes contraires ; ou encore si le produit f (A).f (B) est négatif.

La dichotomie permet de réduire la fourchette jusqu'à ce que $|B - A|$ soit aussi petit que l'on veut. On écrit $|B - A| < PR$(précision).

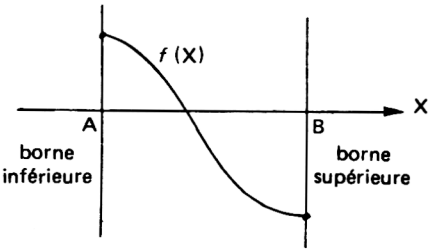


Fig. 2.

Procédé : Calculons $\frac{A + B}{2}$ = nouvelle borne. Deux cas sont possibles.

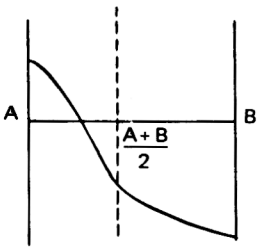


Fig. 3.

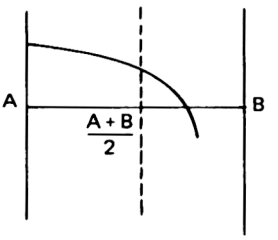


Fig. 4.

– Si $f\left(\frac{A + B}{2}\right) . f(A) < 0$, cas de la figure ci-dessus, la solution se trouve entre A et $\frac{A+B}{2}$. Les bornes sont maintenant A comme inférieure et $\frac{A + B}{2}$ comme supérieure. Au fond, si $f(X).f(A) < 0$ avec $X = \frac{A + B}{2}$, il faut remplacer B par X.

– Si $f\left(\frac{A + B}{2}\right) . f(A) > 0$ la solution se trouve entre la nouvelle borne inférieure $\frac{A + B}{2}$ et l'ancienne borne supérieure B. Si traditionnellement la borne inférieure est désignée par « A », on peut dire que le nouveau « A » est $\frac{A + B}{2}$. Au fond, si $f(X).f(A) > 0$ avec $X = \frac{A + B}{2}$, il faut remplacer A par X.

– On recommencera l'opération jusqu'à obtenir la précision voulue.

Déroulement du programme

Quelques variantes par rapport aux programmes précédents :

Ligne(s)

55	l'ordinateur pose la question concernant la graduation des axes (OUI ou NON)
520 à 560	si B$ = "OUI", on gradue les axes au 1/5 et au 3/5 de N (ligne 56) (Procédure un peu longue)
1000 à 1110	méthode de dichotomie. PR : précision prise ici (au 1/1000)

Remarque importante

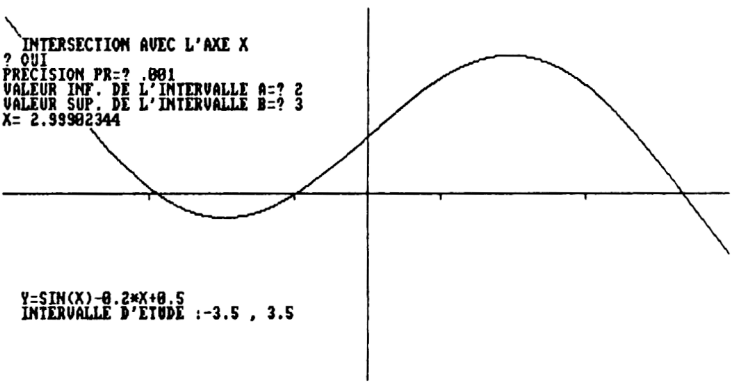
Dans certains cas, il peut se faire que Q = FNY (X) (1060) soit égal à zéro. Rajouter alors en 1065 :

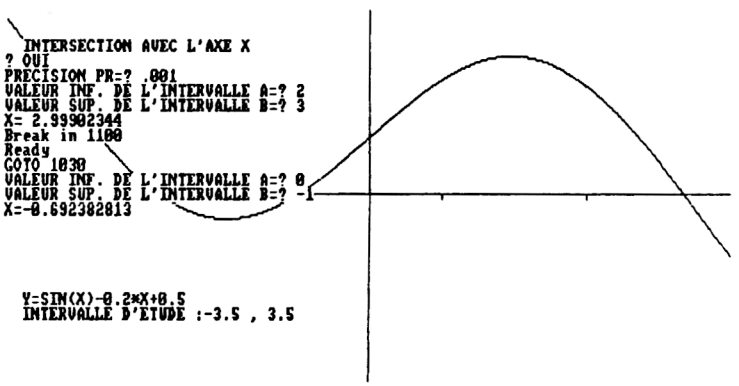
Ligne

1065 IF Q = 0 THEN PRINT "X = " ; X : STOP

```
10 REM FONCTION Y=F(X)
20 REM ***************
30 MODE 2
40 INPUT "ETUDE DE LA FONCTION Y=";A$
50 INPUT "INTERVALLE D'ETUDE N=";N
55 INPUT "GRADUATION DES AXES";B$
56 IF B$="OUI" THEN P=N/5
60 CLS
70 LOCATE 15,12:PRINT "CALCUL EN COURS"
80 DIM A(640):S=0
90 FOR F=1 TO 640
95 LOCATE 3,15:PRINT F
100 X=(F-320)*N/320
120 DEF FNY(X)=SIN(X)-0.2*X+0.5
125 A(F)=FNY(X)
130 IF ABS(A(F))>S THEN S=ABS(A(F))
200 NEXT F
210 CLS
300 REM TRACE DE LA FONCTION
310 REM ********************
320 PLOT 320,0:DRAWR 0,400
330 PLOT 0,200:DRAWR 640,0
340 T=0:IF B$="OUI" THEN GOSUB 520
350 FOR F=1 TO 640
360 IF ABS(A(F))/S=1 THEN T=0:GOTO 400
370 Y=A(F)/S*190
380 IF T=0 THEN PLOT F,Y+200:T=1:GOTO 400
390 DRAWR 1,Y-AY
400 AY=CINT(A(F)/S*190)
410 NEXT F
420 LOCATE 3,20:PRINT "Y=";A$
430 LOCATE 3,21:PRINT "INTERVALLE D'ETUDE :";-N;",
";N
```

```
450 LOCATE 3,3:PRINT "INTERSECTION AVEC L'AXE X":I
NPUT C$
460 IF C$="OUI" THEN GOTO 1020 ELSE STOP
500 REM GRADUATION DES AXES
510 REM ******************
520 FOR F=1 TO 640
530 FOR L=-N TO N STEP P
540 IF F=CINT(640*L/N) THEN PLOT F,200:DRAWR 0,-5
550 NEXT L
560 NEXT F
570 RETURN
1000 REM DICHOTOMIE
1010 REM **********
1020 INPUT "PRECISION PR=";PR
1030 INPUT "VALEUR INF. DE L'INTERVALLE A=";A
1040 INPUT "VALEUR SUP. DE L'INTERVALLE B=";B
1050 X=(A+B)/2
1060 Q=FNY(X):R=FNY(A)
1070 Z=Q*R
1080 IF Z>0 THEN A=X
1090 IF Z<0 THEN B=X
1100 IF ABS(B-A)<PR THEN PRINT "X=";X:STOP
1110 GOTO 1050
```





Programme 4

Ce programme va vous permettre de faire apparaître sur l'écran, si vous le désirez, la dérivée $Y' = f(X)$ de la fonction $Y = f(X)$.

Remarque importante : Dans les programmes 3 et 4, n'oubliez pas d'écrire en ligne 110 les conditions particulières à l'étude de la fonction (intervalle de définition, etc.).

La dérivée est calculée numériquement à droite (ligne 140) et à gauche (ligne 150) de chaque point considéré et ensuite moyennée (ligne 160).

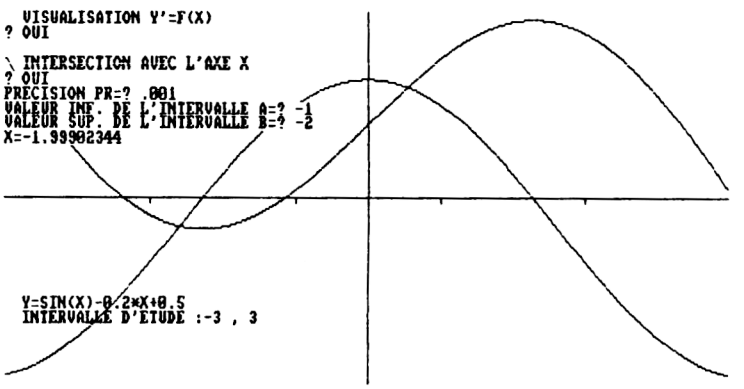
Ligne(s)

170 et 180	calcul de l'échelle automatique pour visualiser la courbe et sa dérivée
440	l'ordinateur demande à l'utilisateur s'il veut visualiser la dérivée $Y' = f(X)$
450	l'ordinateur demande à l'utilisateur le calcul de l'intersection avec l'axe des X (dichotomie)

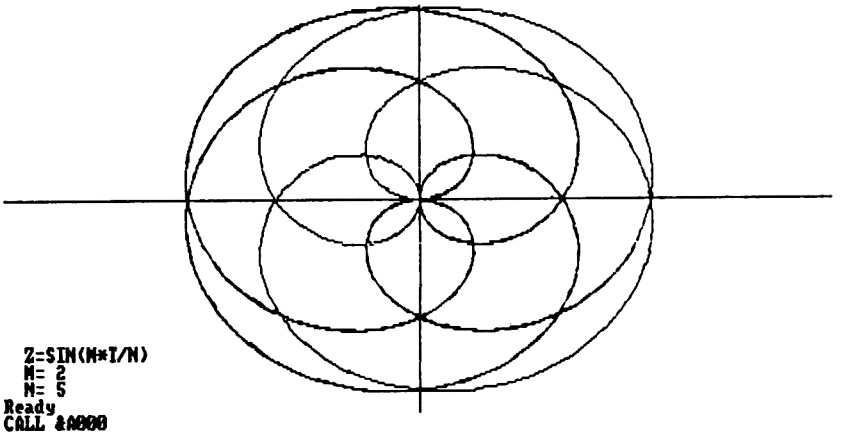
Les programmes 5, 6, 7, 8 et 9 présentent des études de courbes en coordonnées polaires. Les méthodes de calcul sont identiques aux programmes 1 et 2.

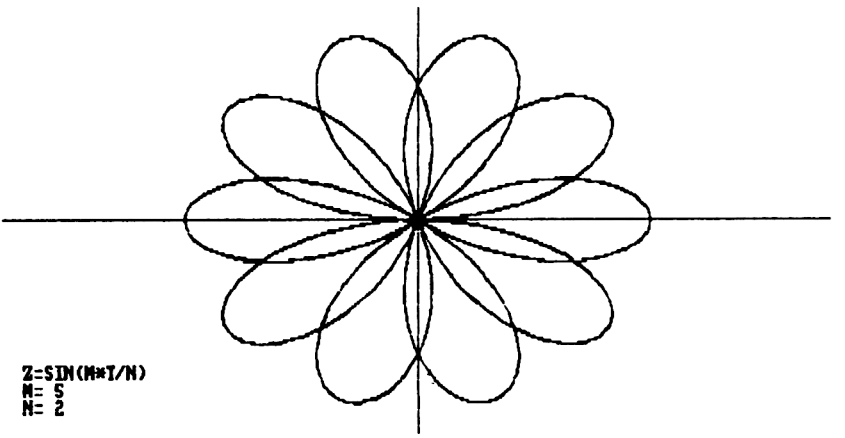
```
10 REM FONCTION Y=F(X)
20 REM ***************
30 MODE 2
40 INPUT "ETUDE DE LA FONCTION Y=";A$
50 INPUT "INTERVALLE D'ETUDE N=";N
55 INPUT "GRADUATION DES AXES";B$
56 IF B$="OUI" THEN P=N/5
60 CLS
70 LOCATE 15,12:PRINT "CALCUL EN COURS"
80 DIM A(640):DIM D(640):S=0:J=0:DX=0.001
90 FOR F=1 TO 640
95 LOCATE 3,15:PRINT F
100 X=(F-320)*N/320:A=X
120 DEF FNY(X)=SIN(X)-0.2*X+0.5
130 A(F)=FNY(X)
140 X=A+DX:Z1=FNY(X)
150 X=A-DX:Z2=FNY(X)
160 D(F)=(Z1-Z2)/(2*DX)
170 IF ABS(A(F))>S THEN S=ABS(A(F))
180 IF ABS(D(F))>J THEN J=ABS(D(F))
200 NEXT F
210 CLS
300 REM TRACE DE LA FONCTION
310 REM ********************
320 PLOT 320,0:DRAWR 0,400
330 PLOT 0,200:DRAWR 640,0
340 T=0:IF B$="OUI" THEN GOSUB 520
350 FOR F=1 TO 640
```

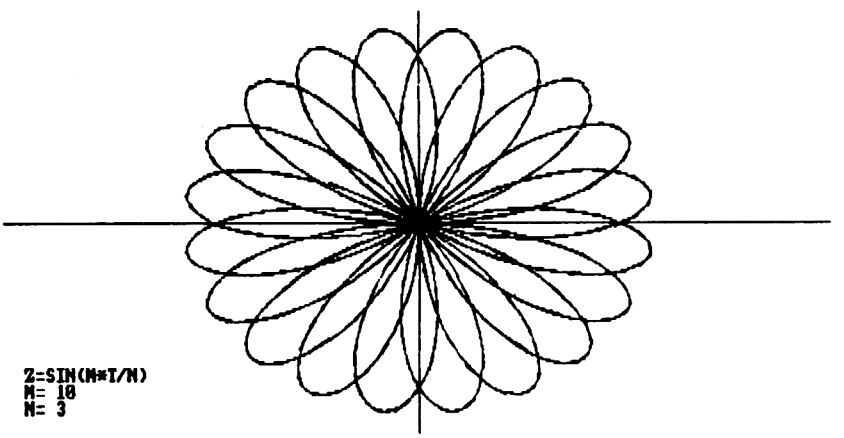
```
360 IF ABS(A(F))/S=1 THEN T=0:GOTO 400
370 Y=A(F)/S*190
380 IF T=0 THEN PLOT F,Y+200:T=1:GOTO 400
390 DRAWR 1,Y-AY
400 AY=CINT(A(F)/S*190)
410 NEXT F
420 LOCATE 3,20:PRINT "Y=";A$
430 LOCATE 3,21:PRINT "INTERVALLE D'ETUDE :";-N;",";N
440 LOCATE 3,1:PRINT "VISUALISATION Y'=F(X)":INPUT D$
450 LOCATE 3,4:PRINT "INTERSECTION AVEC L'AXE X":INPUT C
455 IF D$="OUI" THEN GOTO 620
460 IF C$="OUI" THEN GOTO 1020 ELSE STOP
500 REM GRADUATION DES AXES
510 REM ******************
520 FOR F=1 TO 640
530 FOR L=-N TO N STEP P
540 IF F=CINT(640*L/N) THEN PLOT F,200:DRAWR 0,-5
550 NEXT L
560 NEXT F
570 RETURN
600 REM DERIVEE Y'=F(X)
610 REM *************
620 T=0
630 FOR F=1 TO 640
640 IF ABS(D(F))/J=1 THEN T=0:GOTO 700
650 D=D(F)/J*190
660 IF T=0 THEN PLOT F,D+200:T=1:GOTO 700
670 DRAWR 1,D-AD
700 AD=CINT(D(F)/J*190)
710 NEXT F
800 IF C$="NON" THEN STOP
1000 REM DICHOTOMIE
1010 REM **********
1020 INPUT "PRECISION PR=";PR
1030 INPUT "VALEUR INF. DE L'INTERVALLE A=";A
1040 INPUT "VALEUR SUP. DE L'INTERVALLE B=";B
1050 X=(A+B)/2
1060 Q=FNY(X):R=FNY(A)
1070 Z=Q*R
1080 IF Z>0 THEN A=X
1090 IF Z<0 THEN B=X
1100 IF ABS(B-A)<PR THEN PRINT "X=";X:STOP
1110 GOTO 1050
```



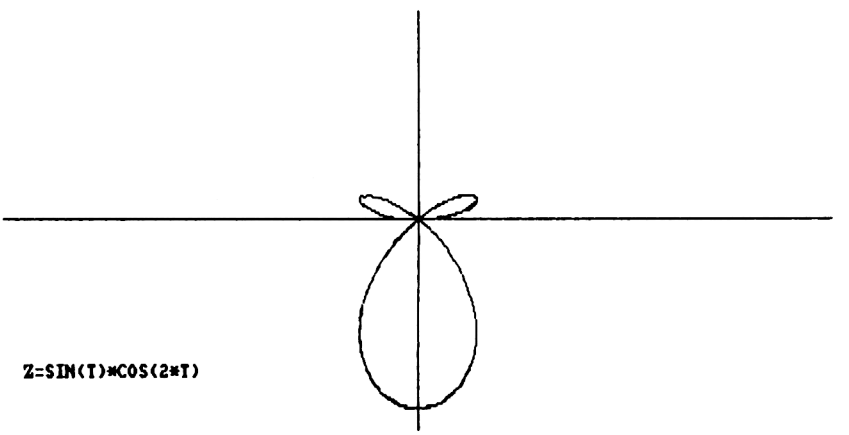
```
10 REM COURBE:COORDONNEES POLAIRES
20 REM ***************************
30 REM
40 CLS: MODE 2
50 INPUT "ETUDE DE LA FONCTION:Z=";Z$
55 INPUT "VALEUR DE M=";M
56 INPUT "VALEUR DE N=";N
60 LOCATE 20,12:PRINT "CALCUL EN COURS"
70 DIM A(800):DIM X(800):DIM Y(800)
80 CX=0:CY=0:S=0
90 FOR F=1 TO 800
100 T=50*F/800
110 LOCATE 3,15:PRINT "T=";CINT(T)
120 DEF FNY(T)=SIN(M*T/N)
130 A(F)=FNY(T):X(F)=A(F)*COS(T):Y(F)=A(F)*SIN(T)
140 IF ABS(X(F))>CX THEN CX=ABS(X(F))
150 IF ABS(Y(F))>CY THEN CY=ABS(Y(F))
155 NEXT F
160 IF CX>CY THEN S=CX
170 IF CY>CX THEN S=CY
175 CLS
180 REM TRACE DE LA COURBE
190 REM ******************
200 PLOT 320,0:DRAWR 0,400
210 PLOT 0,200:DRAWR 640,0
220 FOR F=1 TO 800
230 IF F=800 THEN GOTO 400
240 PLOT X(F)/S*180+320,Y(F)/S*180+200:DRAWR(X(F+1)-X(F))/S*180,
(Y(F+1)-Y(F))/S*180
250 NEXT F
400 LOCATE 3,22:PRINT "Z=";Z$
410 LOCATE 3,23:PRINT "M=";M
420 LOCATE 3,24:PRINT "N=";N
```


Z=SIN(M*T/N)
M= 2
N= 5
Ready
CALL &A000


Z=SIN(M*T/N)
M= 5
N= 2

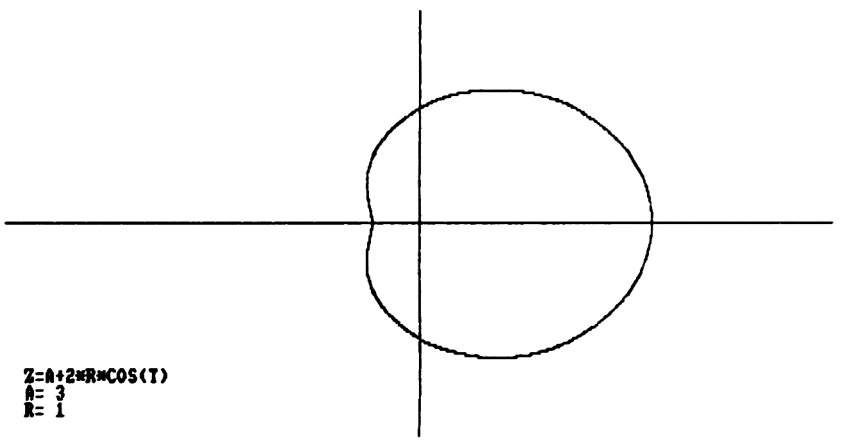

Z=SIN(M*T/N)
M= 10
N= 3

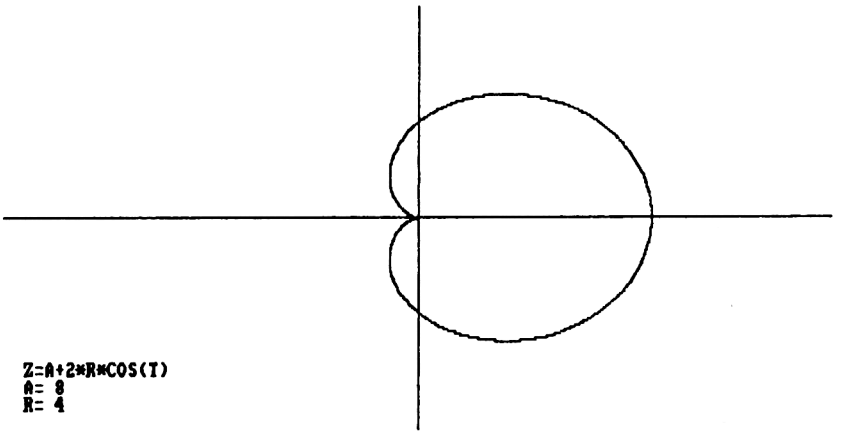
```
10 REM COURBE:COORDONNEES POLAIRES
20 REM ***************************
30 REM
40 CLS: MODE 2
50 INPUT "ETUDE DE LA FONCTION:Z=";Z$
60 LOCATE 20,12:PRINT "CALCUL EN COURS"
70 DIM A(90):DIM X(90):DIM Y(90)
80 CX=0:CY=0:S=0
90 FOR F=1 TO 90
100 T=5*F/90
110 LOCATE 3,15:PRINT "T=";CINT(T)
120 DEF FNY(T)=SIN(T)*COS(2*T)
130 A(F)=FNY(T):X(F)=A(F)*COS(T):Y(F)=A(F)*SIN(T)
140 IF ABS(X(F))>CX THEN CX=ABS(X(F))
150 IF ABS(Y(F))>CY THEN CY=ABS(Y(F))
155 NEXT F
160 IF CX>CY THEN S=CX
170 IF CY>CX THEN S=CY
175 CLS
180 REM TRACE DE LA COURBE
190 REM ******************
200 PLOT 320,0:DRAWR 0,400
210 PLOT 0,200:DRAWR 640,0
220 FOR F=1 TO 90
230 IF F=90 THEN GOTO 400
240 PLOT X(F)/S*180+320,Y(F)/S*180+200:DRAWR (X(F+
1)-X(F))/S*180,(Y(F+1)-Y(F))/S*180
250 NEXT F
400 LOCATE 3,22:PRINT "Z=";Z$
```



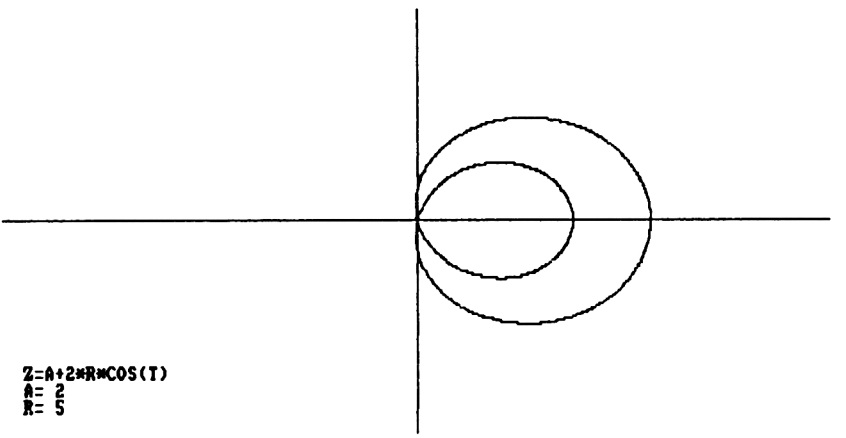
Z=SIN(T)*COS(2*T)

```
10 REM COURBE:COORDONNEES POLAIRES
20 REM ***************************
30 REM
40 CLS: MODE 2
50 INPUT "ETUDE DE LA FONCTION:Z=";Z$
55 INPUT "VALEUR DE A=";A
56 INPUT "VALEUR DE R=";R
60 LOCATE 20,12:PRINT "CALCUL EN COURS"
70 DIM A(800):DIM X(800):DIM Y(800)
80 CX=0:CY=0:S=0
90 FOR F=1 TO 800
100 T=2*PI*F/800
110 LOCATE 3,15:PRINT "T=";CINT(T)
120 DEF FNY(T)=A+2*R*COS(T)
130 A(F)=FNY(T):X(F)=A(F)*COS(T):Y(F)=A(F)*SIN(T)
140 IF ABS(X(F))>CX THEN CX=ABS(X(F))
150 IF ABS(Y(F))>CY THEN CY=ABS(Y(F))
155 NEXT F
160 IF CX>CY THEN S=CX
170 IF CY>CX THEN S=CY
175 CLS
180 REM TRACE DE LA COURBE
190 REM *****************
200 PLOT 320,0:DRAWR 0,400
210 PLOT 0,200:DRAWR 640,0
220 FOR F=1 TO 800
230 IF F=800 THEN GOTO 400
240 PLOT X(F)/S*180+320,Y(F)/S*180+200:DRAWR (X(F+
1)-X(F))/S*180,(Y(F+1)-Y(F))/S*180
250 NEXT F
400 LOCATE 3,22:PRINT "Z=";Z$
410 LOCATE 3,23:PRINT "A=";A
420 LOCATE 3,24:PRINT "R=";R
```

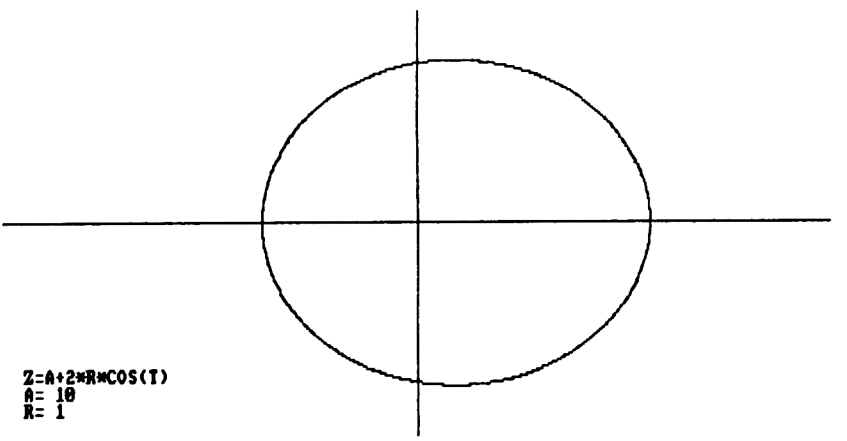




```
Z=A+2*R*COS(T)
A= 8
R= 4
```

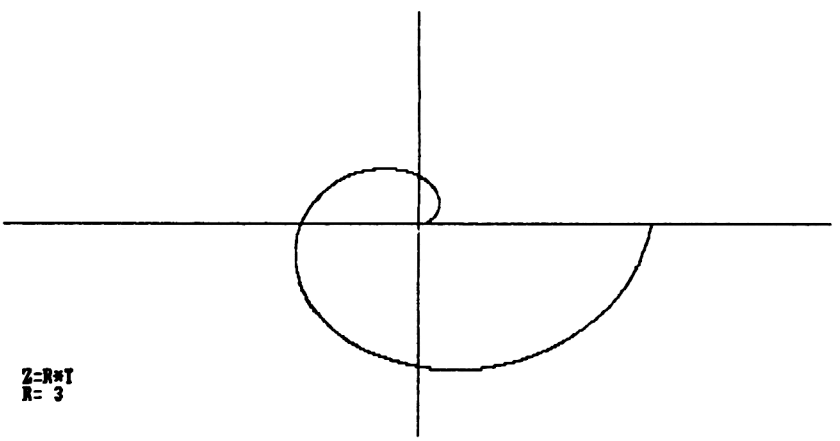


```
Z=A+2*R*COS(T)
A= 2
R= 5
```

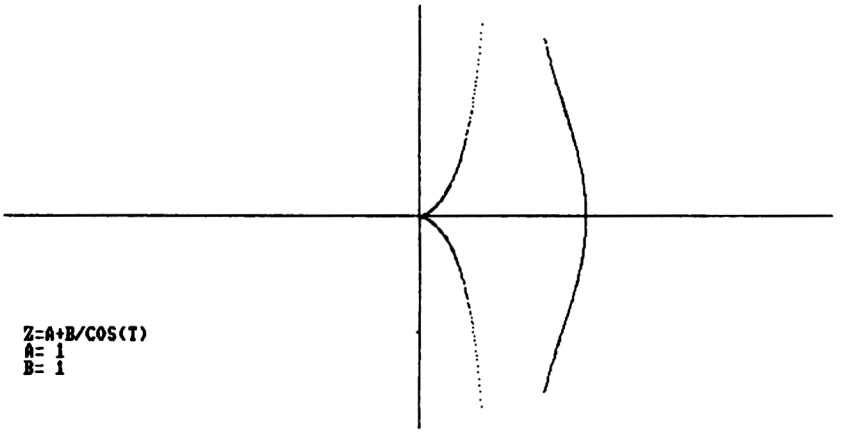


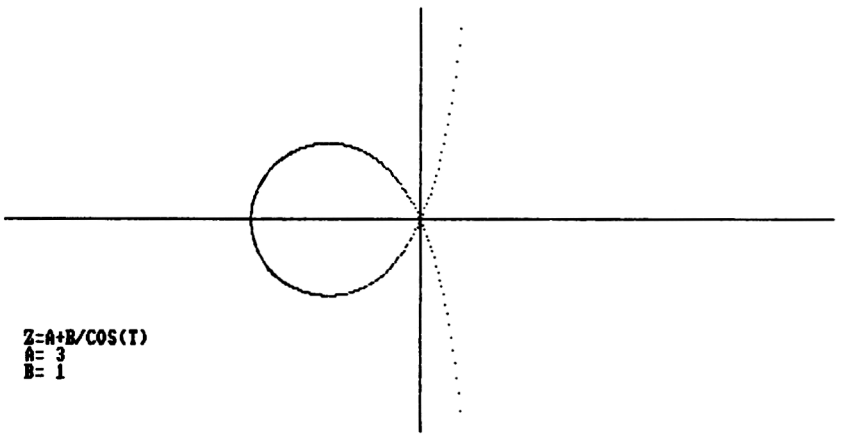
```
Z=A+2*R*COS(T)
A= 10
R= 1
```

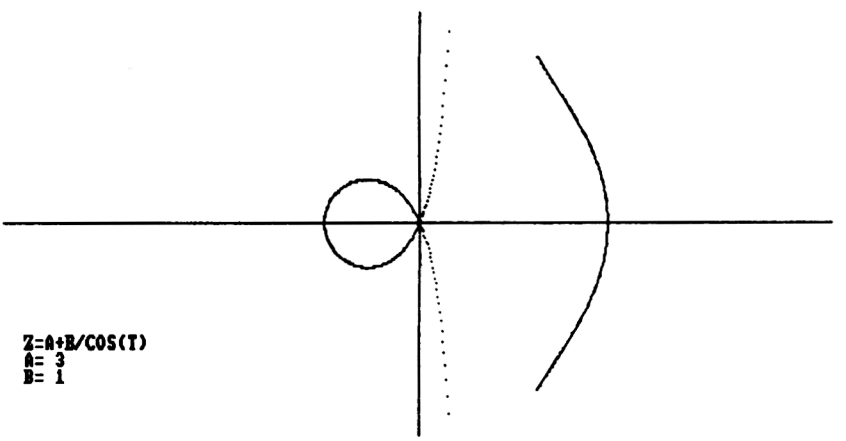
```
10 REM COURBE:COORDONNEES POLAIRES
20 REM ***************************
30 REM
40 CLS: MODE 2
50 INPUT "ETUDE DE LA FONCTION:Z=";Z$
55 INPUT "VALEUR DE R=";R
60 LOCATE 20,12:PRINT "CALCUL EN COURS"
70 DIM A(800):DIM X(800):DIM Y(800)
80 CX=0:CY=0:S=0
90 FOR F=1 TO 800
100 T=2*PI*F/800
110 LOCATE 3,15:PRINT "T=";CINT(T)
120 DEF FNY(T)=R*T
130 A(F)=FNY(T):X(F)=A(F)*COS(T):Y(F)=A(F)*SIN(T)
140 IF ABS(X(F))>CX THEN CX=ABS(X(F))
150 IF ABS(Y(F))>CY THEN CY=ABS(Y(F))
155 NEXT F
160 IF CX>CY THEN S=CX
170 IF CY>CX THEN S=CY
175 CLS
180 REM TRACE DE LA COURBE
190 REM *****************
200 PLOT 320,0:DRAWR 0,400
210 PLOT 0,200:DRAWR 640,0
220 FOR F=1 TO 800
230 IF F=800 THEN GOTO 400
240 PLOT X(F)/S*180+320,Y(F)/S*180+200:DRAWR (X(F+
1)-X(F))/S*180,(Y(F+1)-Y(F))/S*180
250 NEXT F
400 LOCATE 3,22:PRINT "Z=";Z$
410 LOCATE 3,23:PRINT "R=";R
```



```
10 REM COURBE:COORDONNEES POLAIRES
20 REM ***************************
30 REM
40 CLS: MODE 2
50 INPUT "ETUDE DE LA FONCTION:Z=";Z$
55 INPUT "VALEUR DE A=";A
56 INPUT "VALEUR DE B=";B
60 LOCATE 20,12:PRINT "CALCUL EN COURS"
70 DIM A(800):DIM X(800):DIM Y(800)
80 CX=0:CY=0:S=0
90 FOR F=1 TO 800
100 T=2*PI*F/800
105 IF COS(T)=0 THEN A(F)=0:GOTO 155
110 LOCATE 3,15:PRINT "T=";CINT(T)
120 DEF FNY(T)=A+B/COS(T)
125 A(F)=FNY(T)
126 IF ABS(A(F))>3 THEN A(F)=0:X(F)=0:Y(F)=0:GOTO
155
130 X(F)=A(F)*COS(T):Y(F)=A(F)*SIN(T)
140 IF ABS(X(F))>CX THEN CX=ABS(X(F))
150 IF ABS(Y(F))>CY THEN CY=ABS(Y(F))
155 NEXT F
160 IF CX>CY THEN S=CX
170 IF CY>CX THEN S=CY
175 CLS
180 REM TRACE DE LA COURBE
190 REM ******************
200 PLOT 320,0:DRAWR 0,400
210 PLOT 0,200:DRAWR 640,0
220 FOR F=1 TO 800
225 IF A(F)=0 THEN GOTO 250
230 IF F=800 THEN GOTO 400
240 PLOT X(F)/S*180+320,Y(F)/S*180+200
250 NEXT F
400 LOCATE 3,20:PRINT "Z=";Z$
410 LOCATE 3,21:PRINT "A=";A
420 LOCATE 3,22:PRINT "B=";B
```


Z=A+B/COS(T)
A= 1
B= 1


Z=A+B/COS(T)
A= 3
B= 1


Z=A+B/COS(T)
A= 3
B= 1

Primitives et équations différentielles

L'utilisateur propose une fonction y (x), l'ordinateur trace les graphes de y (x) et de deux primitives successives. Telle est l'idée première de ce programme, mais il permet également de tracer le graphe de la solution d'une équation différentielle du second ordre.

Déroulement du programme

La fonction y (x) est portée en ligne 1000. La fonction sera étudiée dans l'intervalle O, X. Dans cet intervalle, elle sera calculée pour 100 valeurs de x (ligne 270).

On peut éviter de recopier les lignes 10 à 40 qui ne constituent qu'un préambule.

Pour lancer le programme lui-même, on fait RUN 200.

La primitive de y est notée z (x), la primitive de z est notée t (x). L'utilisateur fixe X, z (O) et t (O).

Pour chaque valeur de x, une valeur approchée de z (x) est calculée en ajoutant y $(x).dx$ à la valeur précédente. L'ordinateur calcule de même t (x). Ces valeurs sont mémorisées.

De plus l'ordinateur cherche si les valeurs de y, z et de t sont supérieures ou inférieures aux valeurs précédentes. Ceci afin de déterminer l'échelle e en ordonnée de la représentation des graphes.

Enfin il trace les courbes représentant y (x), z (x) et t (x).

Critique

Il est bien évident que le pas d'intégration dx n'est pas infiniment petit et que donc les résultats seront approchés. Tel que le programme est écrit, dx est représenté par 6 pixels sur l'écran.

Vous pouvez le réduire à un pixel soit $dx = X/600$ en remplaçant 100 par 600, 98 par 598, 6 par 1 en lignes 210, 270, 520 et 530. Vous pouvez également prendre un dx encore cent fois plus petit, mais en ne marquant qu'un point sur cent car un marquage plus serré serait inutile, et en ne mémorisant les valeurs de y qu'une fois sur cent pour ne pas surcharger la mémoire.

Tout ceci améliore la précision du calcul mais augmente le temps nécessaire à la réalisation du programme... à vous de choisir.

Détail du programme

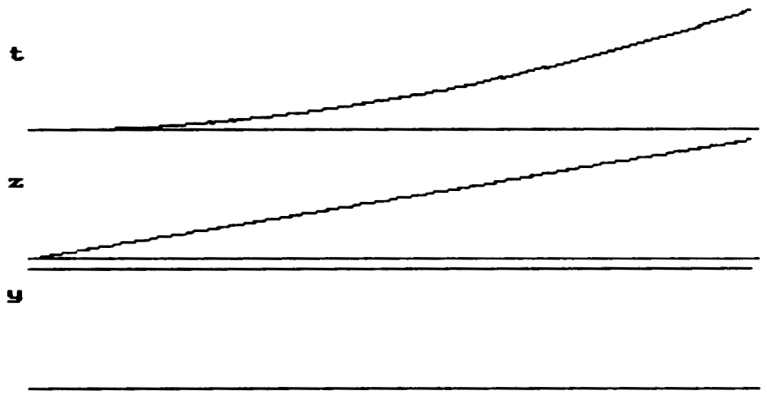
Ligne(s)	
10	présentation
1000	fonction y (x)
200	introduction des données
280	situation pour $x = 0$
320	boucle de calcul
340	calcul de y (x)
360	calcul de z (x)
370	calcul de t (x)
420	détermination des échelles
470	représentation graphique

Liste des variables

x variable
y $(0,x)$ valeur de y (x)
y $(1,x)$ valeur de z (x)
y $(2,x)$ valeur de t (x)
f valeur maximale de x
dx accroissement de x entre deux calculs
k indice indiquant y ou z ou t
n numéro du calcul $(x = n.dx)$
p (k) valeur du maximum de y, z ou t
m (k) valeur du minimum de y ou z ou t
e (k) échelle en ordonnée de y, z ou t
b (k) décalage en ordonnée pour y, z ou t afin de placer l'axe des abscisses.

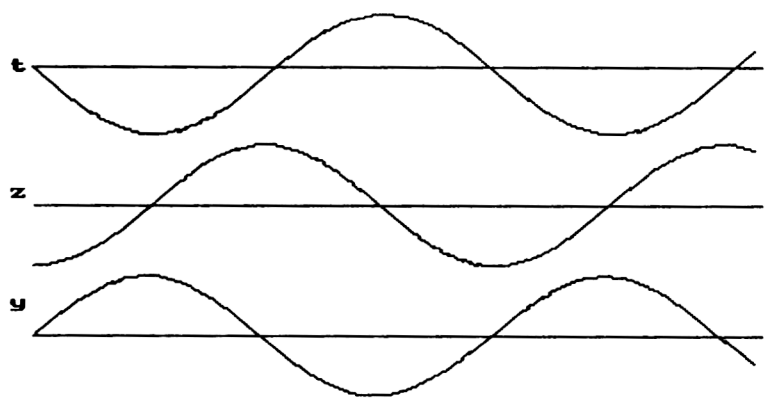
Illustrations

Figure 1



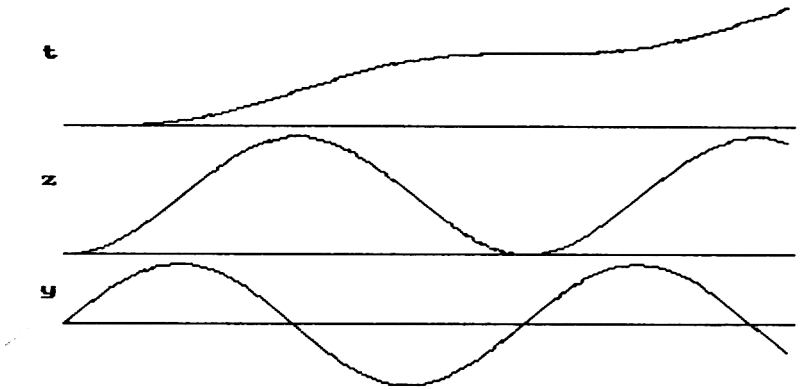
$y = 4$ avec $z\ (0) = 0$, $t\ (0) = 0$ et $X = 10$
la théorie prévoit $z = 4.x$ et $t = 8.x$

Figure 2



$y = \sin\ (x)$ avec $z\ (0) = -1$, $t\ (0) = 0$ et $X = 10$
la théorie prévoit $z = -\cos\ (x)$ et $t = -\sin\ (x)$

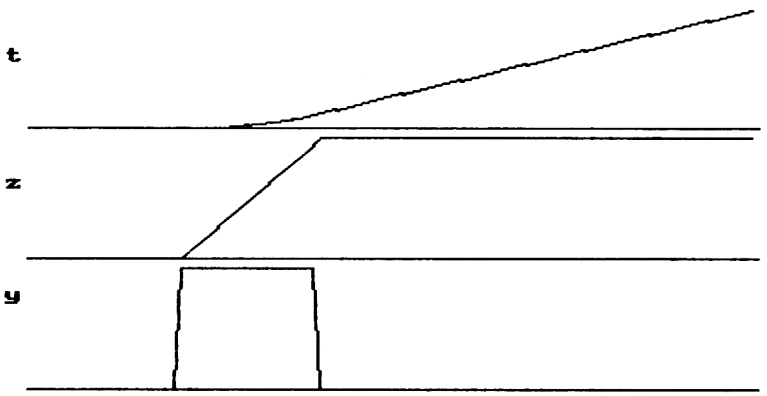
Figure 3



$y = \sin(x)$ avec $z(0) = 0$, $t(0) = 0$ et $X = 10$

la théorie prévoit $z = -\cos(x) + 1$ et $t = -\sin(x) + x$

Figure 4

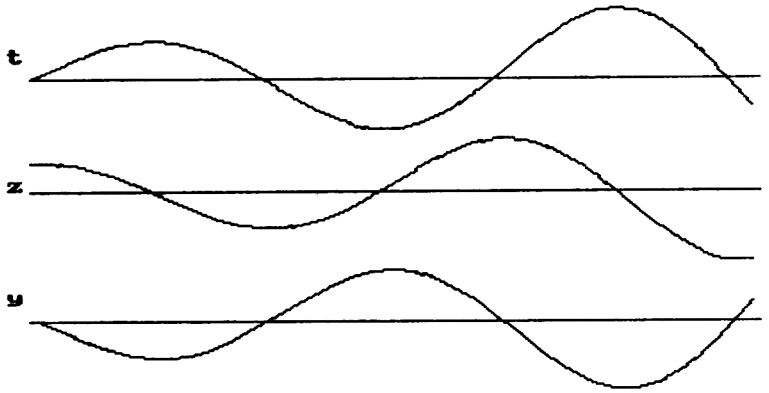


IF X > 10 AND X < 20 THEN Y = 1 ELSE Y = 0

avec $z(0) = 0$, $t(0) = 0$ et $X = 50$

Dans tous ces cas, l'ordinateur fournit des réponses satisfaisantes.

Figure 5



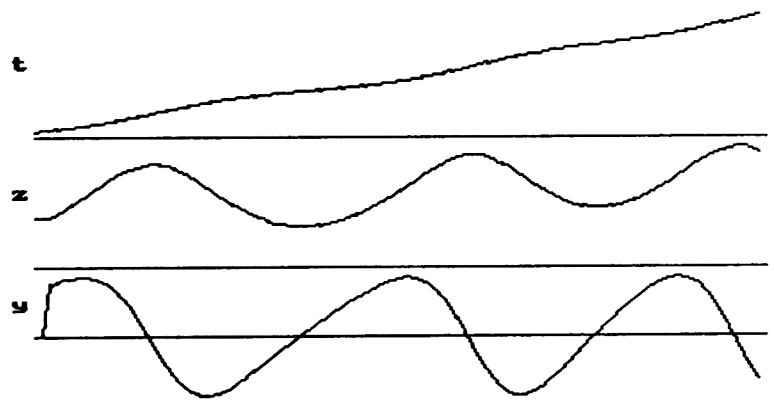
Résolvons l'équation différentielle $\dfrac{d^2\, t\,(x)}{dx^2} = -t\,(x)$

soit $y = -t$

avec $z\,(0) = 1$, $t\,(0) = 0$ et $X = 10$

On s'attend à observer $t\,(x) = \sin\,(x)$ mais on note une divergence dans la réponse. C'est que le dx égal à X/100 soit 0,1 n'est pas très petit devant la période de sin (x) qui est égale à 2.PI. Le calcul est donc approché et devient très mauvais si l'on prend par exemple X = 100.

Figure 6



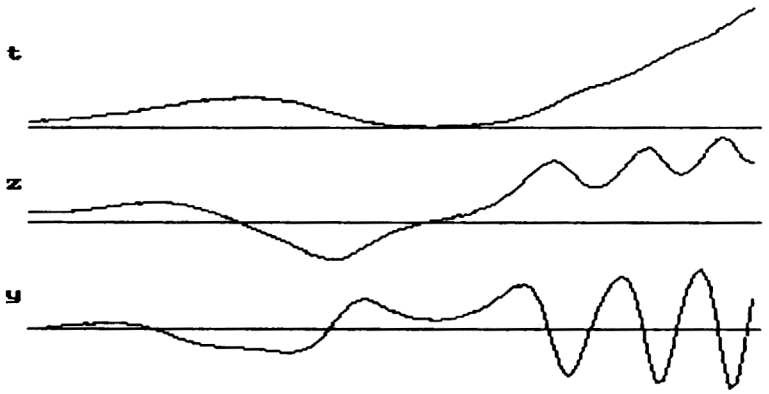
Résolvons l'équation différentielle $\dfrac{d^2\, t\,(x)}{dx^2} = \sin\,(t\,(x))$

soit $y = \sin\,(t)$ avec $z\,(0) = 1$, $t\,(0) = .1$ et $X = 10$.

On peut s'interroger sur le début de la courbe. Essayons alors X = 1 : le défaut a approximativement la même largeur, il est donc du au fait que notre procédé de calcul impose $y\,(0) = 0$ ce qui ne

correspond pas forcément à la solution de l'équation différentielle. Il y a donc au départ de la courbe régime transitoire entre cette solution forcée et la solution de l'équation différentielle.

Figure 7



Résolvons l'équation $\frac{d^2\, t\,(x)}{dx^2} = x\,.\sin\,(t\,(x))$

soit $y = x\,.\sin\,(t)$ avec $z\,(0) = 1$, $t\,(0) = 1$ et $X = 10$

Quel plaisir de pouvoir observer aussi rapidement la forme de cette équation différentielle !

Bien entendu toutes ces familles de courbes peuvent être utilisées comme exemples d'une fonction et de ses deux premières dérivées. Ceci permet de bien réaliser que la dérivée est liée à la pente de la fonction.

Programme

```
10 CLS
20 LOCATE 14,6:PRINT"primitives"
30 LOCATE 6,8:PRINT"et equations differentielles"
40 LOCATE 1,22:PRINT"pour faire avancer le progra
me, taper "CHR$(243)
50 LOCATE 39,25:PRINT CHR$(243)
60 q=INKEY(1):IF q<>0 THEN 60
70 CLS
80 PRINT:PRINT"x varie de 0 a X (X>0)"
90 PRINT:PRINT"z est la primitive de y"
100 PRINT:PRINT"t est la primitive de z "
110 PRINT:PRINT"donner la fonction y(x,z,t)"
120 PRINT:PRINT"en ligne 1000"
130 PRINT:PRINT"puis faire RUN 200"
```

```
140 LIST 1000
190 REM
200 CLS
210 DIM y(2,100)
220 PRINT:INPUT"X =";f
230 PRINT:INPUT"valeur de z pour x=0 ";y(1,0)
240 PRINT:INPUT"valeur de t pour x=0 ";y(2,0)
250 PRINT:PRINT"calcul en cours"
260 REM
270 dx=f/100
280 x=0:GOSUB 1000:y(0,0)=y
290 FOR k=0 TO 2
300 m(k)=MIN(0,y(k,0)):p(k)=MAX(0,y(k,0))
310 NEXT k
320 FOR n=1 TO 99
330 x=n*dx
340 GOSUB 1000
350 y(0,n)=y:p(0)=MAX(p(0),y):m(0)=MIN(m(0),y)
360 y(1,n)=y(1,n-1)+dx*y(0,n-1)
370 z=y(1,n) :p(1)=MAX(p(1),z):m(1)=MIN(m(1),z)
380 y(2,n)=y(2,n-1)+dx*y(1,n-1)
390 t=y(2,n) :p(2)=MAX(p(2),t):m(2)=MIN(m(2),t)
400 NEXT n
410 REM
420 FOR k=0 TO 2
430 e(k)=120/(p(k)-m(k))
440 b(k)=e(k)*m(k)
450 NEXT k
460 CLS
470 LOCATE 1,4:PRINT"t"
480 LOCATE 1,12:PRINT"z"
490 LOCATE 1,19:PRINT"y"
500 FOR k=0 TO 2
510 MOVE 20,10+130*k-b(k):DRAWR 600,0
520 FOR n=0 TO 98
530 MOVE 20+6*n,10+130*k-b(k)+e(k)*y(k,n)
540 DRAWR 6,e(k)*(y(k,n+1)-y(k,n))
550 NEXT n
560 NEXT k
570 END
580 REM
1100 RETURN
```

Equations différentielles du second ordre, complètes, linéaires et à coefficients constants

Ce programme permet l'étude des équations linéaires, c'est-à-dire celles où d^2x/dt^2, dx/dt et x sont au premier degré, et dont les coefficients sont constants. Ces équations sont les plus utiles et les plus fréquentes dans les applications pratiques.

Les équations étudiées sont de la forme :

$$A \frac{d^2 x}{dt^2} + B \frac{dx}{dt} + Cx + D = E\sin Wt$$

Le programme permet l'étude des oscillations libres de la forme :

$$A \frac{d^2 x}{dt^2} + Cx = 0$$

Les oscillations forcées de la forme :

$$A \frac{d^2 x}{dt^2} + Cx = E\sin Wt$$

On peut aussi introduire une force retardatrice de la forme $B \frac{dx}{dt}$ ou un terme de frottement sec.

Nous rappelons qu'un « frottement sec » se caractérise par une *force constante* de signe opposé à la vitesse.

Nous proposons à l'utilisateur un grand nombre d'exemples, il lui sera évidemment possible de changer les données pour traiter ses exemples particuliers.

La résolution de l'équation différentielle est numérique, on peut donc reconstituer les régimes transitoires.

Liste des variables

XO abscisse initiale
VO vitesse initiale
A coefficient du terme d^2x/dt^2
B coefficient du terme dx/dt
C coefficient du terme x
D introduction d'un terme permettant de simuler un frottement sec
E coefficient du terme $\sin Wt$
DT pas du calcul
N durée de l'observation du phénomène
CX et CT coefficients multiplicatifs de l'abscisse X et du temps T

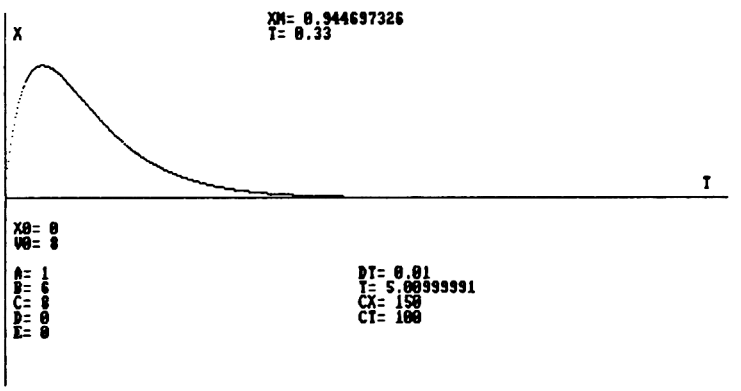
Déroulement du programme

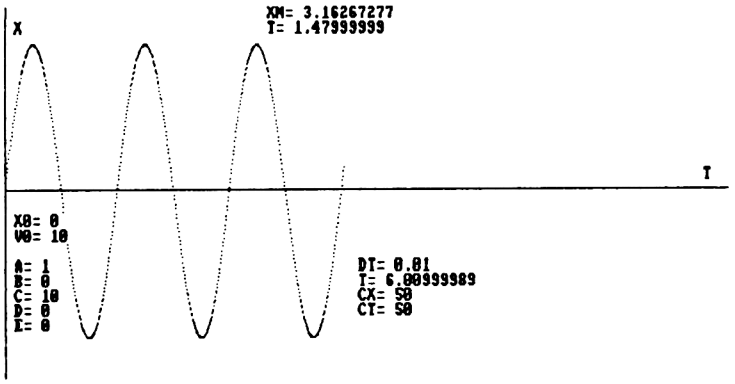
Ligne(s)	
10 à 230	titre et entrées des données
230 - 240	on trace les axes de coordonnées
280 à 310	résolution numérique de l'équation différentielle et on trace la courbe sur l'écran
340	si le coefficient CX choisi par l'utilisateur est trop grand, l'ordinateur avertit en inscrivant la phrase "CX trop grand"
500 à 600	impression sur l'écran des différents paramètres

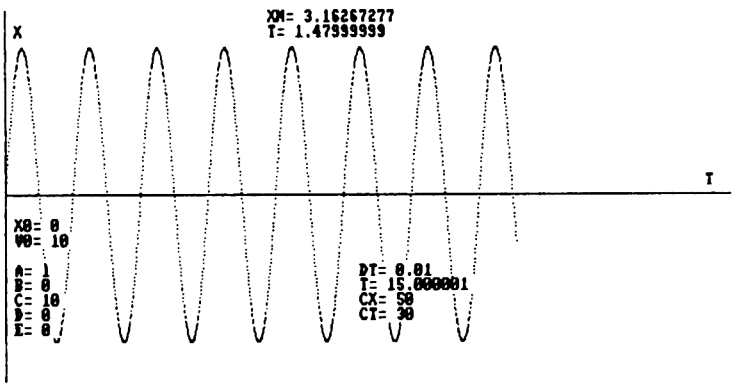
Programme

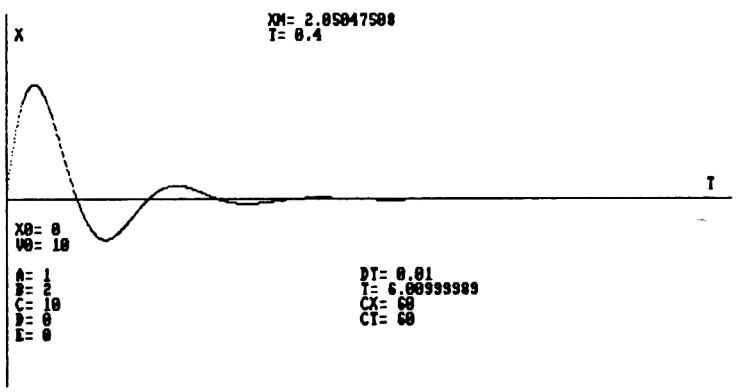
```
10 REM EQUATIONS DIFERENTIELLES
20 REM ************************
30 REM
50 REM EQUATIONS DE LA FORME:
60 REM A*D2X/DT2+B*DX/DT+C*X+D=E*SIN(WT)
70 CLS
80 INPUT "VALEUR DE XO=";XO
90 INPUT "VITESSE INITIALE VO=";VO
100 INPUT "VALEUR DE A=";A
110 INPUT "VALEUR DE B=";B
120 INPUT "VALEUR DE C=";C
130 INPUT "VALEUR DE D=";D
140 INPUT "VALEUR DE E=";E
150 IF E<>0 THEN INPUT "VALEUR DE W=";W
160 IF E=0 THEN W=0
170 INPUT "PAS DU CALCUL DT=";DT
180 INPUT "DUREE DE L'OBSERVATION N=";N
190 INPUT "COEFFICIENT DE X:CX=";CX
200 INPUT "COEFFICIENT DE T:CT=";CT
```

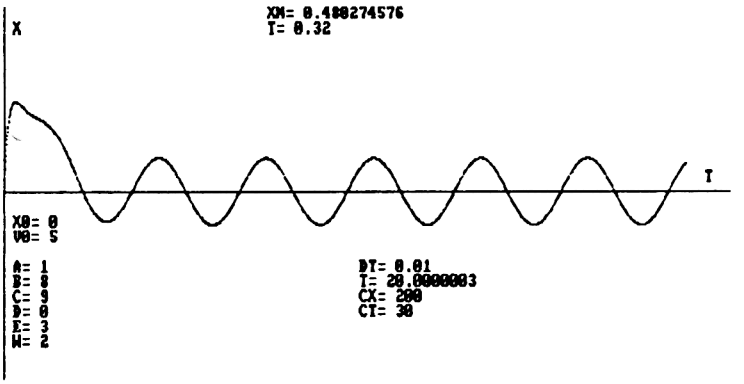
```
220 MODE 2
230 PLOT 0,0:DRAWR 0,400
240 PLOT 0,200:DRAWR 640,0
250 LOCATE 2,2:PRINT "X"
260 LOCATE 78,12:PRINT "T"
270 X=X0:TX=V0:M=ABS(X0)
275 LOCATE 30,1:PRINT "XM=";M:LOCATE 30,2:PRINT "T
=";" 0"
280 FOR T=0 TO N STEP DT
290 ATX=(E*SIN(W*T)-C*X-B*TX-D*SGN(TX))/A
300 ACTX=ATX*DT:TX=TX+ACTX
310 X=X+TX*DT
320 IF ABS(X)>M THEN M=ABS(X):LOCATE 30,1:PRINT "X
M=";M:LOCATE 30,2:PRINT "T=";T
330 IF T*CT>=600 THEN GOTO 500
340 IF X*CX>=180 THEN LOCATE 2,20:PRINT "CX TROP G
RAND":M=0:FOR Z=0 TO 500:NEXT Z:GOTO 190
350 PLOT T*CT,X*CX+200
360 NEXT T
500 LOCATE 2,15:PRINT "X0=";X0
510 LOCATE 2,16:PRINT "V0=";V0
520 LOCATE 2,18:PRINT "A=";A
530 LOCATE 2,19:PRINT "B=";B
540 LOCATE 2,20:PRINT "C=";C
550 LOCATE 2,21:PRINT "D=";D
560 LOCATE 2,22:PRINT "E=";E
565 IF E<>0 THEN LOCATE 2,23:PRINT "W=";W
570 LOCATE 40,18:PRINT "DT=";DT
580 LOCATE 40,19:PRINT "T=";T
590 LOCATE 40,20:PRINT "CX=";CX
600 LOCATE 40,21:PRINT "CT=";CT
```

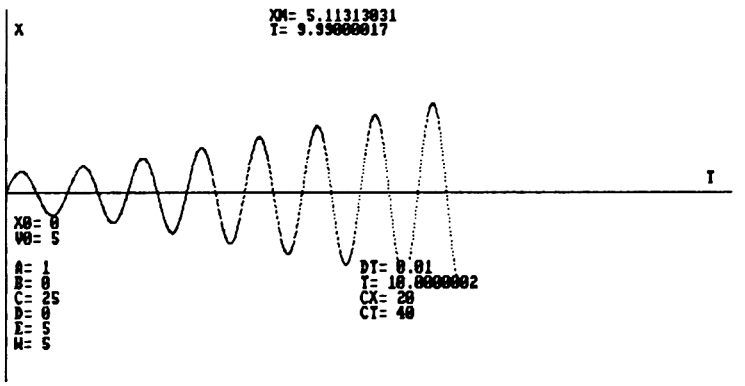


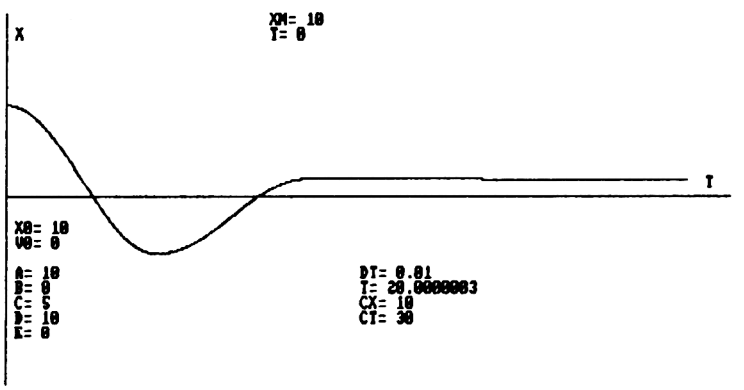

XM= 3.16267277
T= 1.47999999
X
T
X0= 0
V0= 10
A= 1
B= 0
C= 10
D= 0
E= 0
DT= 0.01
T= 6.00999989
CX= 50
CT= 50


XM= 3.16267277
T= 1.47999999
X
T
X0= 0
V0= 10
A= 1
B= 0
C= 10
D= 0
E= 0
DT= 0.01
T= 15.0000001
CX= 50
CT= 30


XM= 2.05047508
T= 0.4
X
T
X0= 0
V0= 10
A= 1
B= 2
C= 10
D= 0
E= 0
DT= 0.01
T= 6.00999989
CX= 60
CT= 60


XM= 0.488274576
T= 0.32
X
T
X0= 0
V0= 5
A= 1
B= 0
C= 9
D= 0
E= 3
W= 2
DT= 0.01
T= 20.0000003
CX= 200
CT= 30


XM= 5.11313831
T= 9.99000017
X
T
X0= 0
V0= 5
A= 1
B= 0
C= 25
D= 0
E= 5
W= 5
DT= 0.01
T= 10.0000002
CX= 20
CT= 40


XM= 10
T= 0
X
T
X0= 10
V0= 0
A= 10
B= 0
C= 5
D= 10
E= 0
DT= 0.01
T= 20.0000003
CX= 10
CT= 30

Méthode de Runge-Kutta

Résolution d'équations différentielles par la méthode de Runge-Kutta

Introduction

Précisons tout de suite qu'il s'agit d'une résolution numérique, ce qui se traduira sur l'écran par une série de nombres. Si, par exemple, l'équation donnée, contient la variable X et la fonction Y (et ses dérivés), on obtiendra des valeurs numériques de Y en face des valeurs numériques de X. Dans toute équation différentielle, apparaissent des conditions initiales X0 et Y0. On donne alors à la variable un incrément H et l'on calcule une moyenne pondérée d'un nombre M d'évaluations, par la formule :

$$Y_{i+1} = Y_i + (K1 + 2\,K2 + 2\,K3 + K4)/6$$

Les coefficients K1, K2, K3, K4 étant déterminés par :

$$* \; K1 = H.F\,(X_i, Y_i)$$
$$* \; K2 = H.F\,(X_i + 1/2\,H, Y_i + 1/2\,K1)$$
$$* \; K3 = H.F\,(X_i + 1/2\,H, Y_i + 1/2\,K2)$$
$$* \; K4 = H.F\,(X_i + H, Y_i + K3)$$

But du programme

Cette méthode s'applique :

1. A un système de deux équations du premier ordre de la forme :

$$Y' = F\,(X, Y, Z)$$
$$Z' = G\,(X, Y, Z)$$

2. Aux équations du second ordre de la forme :

$$Y'' = F\,(X, Y, Y')$$

En effet, il suffit de poser $Y' = Z$, d'où $Y'' = Z'$ pour être ramené au premier cas avec :

$$Y' = Z$$
$$Z' = F\,(X, Y, Z)$$

Déroulement du programme

Ligne(s)

10 à 165 Mise en page et entrée des données
170 à 340 On calcule les coefficients K et on élabore les solutions.
1000 et 1010 Le lecteur doit y inscrire les équations à traiter

Dans un exemple, l'équation différentielle de départ est :

$$Y'' + 4\,Y' + 100\,Y - 100\,X = 0$$

Si l'on pose $Y' = Z$ et $Y'' = Z'$, elle se transforme en :

$$Z' + 4\,Z + 100\,Y - 100\,X = 0$$

ou $$Z' = -4Z - 100\,Y + 100\,X, \text{ avec } Y' = Z.$$

C'est pourquoi, la ligne 1000 contiendra $Y1 = Z$
la ligne 1010 " $Z1 = -4Z - 100\,Y + 100\,X$

Après le RUN, l'ordinateur demande les valeurs initiales.
Par exemple : X0 = 0.1
Y0 = 0.1
Z0 = 0
H = 0.05
M = 9

qui permet d'obtenir 10 valeurs différentes de X et Y.

```
X0= 0.1
Y0= 0.1
Z0= 0
H= 0.05
M= 9

X= 0.15 ,YX= 0.101979167 ,ZX= 0.114479167
X= 0.2 ,YX= 0.114428884 ,ZX= 0.404570694
X= 0.25 ,YX= 0.143779386 ,ZX= 0.774169233
X= 0.3 ,YX= 0.191549802 ,ZX= 1.12693739
X= 0.35 ,YX= 0.254900296 ,ZX= 1.38770729
X= 0.4 ,YX= 0.328050715 ,ZX= 1.51531458
X= 0.45 ,YX= 0.404094366 ,ZX= 1.50557032
X= 0.5 ,YX= 0.476733256 ,ZX= 1.38549684
X= 0.55 ,YX= 0.541564272 ,ZX= 1.20168693
X= 0.6 ,YX= 0.596712437 ,ZX= 1.00643106


Y1=Z
Z1=-4*Z-100*Y+100*X
Ready
CALL &A000
```

Autre exemple Retaper la ligne 1000 : Y1 = EXP (X) — Y
Taper en 1010 : Z1 = 0
Faites un RUN. Entrer X0 = 0, Y0 = 0, Z0 = 0, H = 0.5, M = 4, par exemple.
On a résolu cette fois l'équation du premier ordre Y' = EXP (X) — Y.

```
X0= 0
Y0= 0
Z0= 0
H= 0.5
M= 4

X= 0.5 ,YX= 0.521254225 ,ZX= 0
X= 1 ,YX= 1.17568479 ,ZX= 0
X= 1.5 ,YX= 2.13028713 ,ZX= 0
X= 2 ,YX= 3.62869546 ,ZX= 0
X= 2.5 ,YX= 6.05336328 ,ZX= 0


Y1=EXP(X)-Y
Z1=0
Ready
CALL &A000
```

```
10 REM METHODE DE RUNGE-KUTTA
20 REM **********************
30 CLS
40 REM CONDITIONS INITIALES
50 REM ********************
60 INPUT "VALEUR DE X0=";X0
70 INPUT "VALEUR DE Y0=";Y0
80 INPUT "VALEUR DE Z0=";Z0
90 INPUT "VALEUR DE L'INCREMENT H=";H
100 INPUT "NOMBRE D'EVALUATIONS M=";M
115 MODE 2
120 LOCATE 1,1:PRINT "X0=";X0
130 LOCATE 1,2:PRINT "Y0=";Y0
140 LOCATE 1,3:PRINT "Z0=";Z0
150 LOCATE 1,4:PRINT "H=";H
160 LOCATE 1,5:PRINT "M=";M
165 PRINT
170 FOR N=0 TO M
180 X=X0:Y=Y0:Z=Z0:GOSUB 1000
190 K1=H*Y1:L1=H*Z1
195 X=X0+H/2:Y=Y0+K1/2:Z=Z0+L1/2
200 GOSUB 1000
210 K2=H*Y1:L2=H*Z1
220 Y=Y0+K2/2:Z=Z0+L2/2
230 GOSUB 1000
240 K3=H*Y1:L3=H*Z1
250 X=X0+H:Y=Y0+K3:Z=Z0+L3
```

```
260 GOSUB 1000
270 K4=H*Y1:L4=H*Z1
280 YX=Y0+(K1+2*K2+2*K3+K4)/6
290 ZX=Z0+(L1+2*L2+2*L3+L4)/6
300 PRINT"X=";X;",";"YX=";YX;",";"ZX=";ZX
330 X0=X:Y0=YX:Z0=ZX
340 NEXT N
350 GOTO 1050
500 REM SYST.EQUAT.(LIGNES 1000,1010)
510 REM ****************************
1000 Y1=EXP(X)-Y
1010 Z1=0
1020 V$="EXP(X)-Y
1030 W$="0"
1040 RETURN
1050 LOCATE 1,20:PRINT "Y1=";V$
1060 LOCATE 1,21:PRINT "Z1=";W$
```

Spirale de Cornu

Problème

Soit à construire une courbe dont la courbure $\frac{1}{R}$ en un point soit proportionnelle à la distance S de ce point à un point O pris pour origine des arcs (distance comptée sur la courbe : arc S). On dit que :

$$X = A \int_0^V \cos \frac{\pi V^2}{2} \, dV$$

et $$Y = A \int_0^V \sin \frac{\pi V^2}{2} \, dV$$

En effet, posons : $\pi V^2 = 2 U$

$$X = \int_0^V \frac{A}{\pi V} \cos U dU$$

$$Y = \int_0^V \frac{A}{\pi V} \sin U dU$$

Donc le rayon de courbure $R = \frac{A}{\pi V}$ et l'arc $dS = R$; $dU = AdV$; $S = A.V.$

Enfin $$R = \frac{\pi}{A^2} S$$

La courbe cherchée est donc définie par les intégrales de Fresnel :

$$X = A \int_0^V \cos \frac{\pi V^2}{2} dV \qquad Y = A \int_0^V \sin \frac{\pi V^2}{2} dV$$

Nous rencontrons une difficulté : celle de définir des coordonnées par des intégrales. Pour la tourner, dérivons :

$$\frac{dX}{dU} = A \cos \frac{\pi V^2}{2}$$

$$\frac{dY}{dU} = A \sin \frac{\pi V^2}{2}$$

Système de deux équations différentielles que la méthode de Runge-Kutta nous permet de traiter. Pour la concordance des variables avec le programme, nous remplaçons V par X
X par Y
Y par Z

$$\frac{dY}{dX} = 2 \cos \frac{\pi X^2}{2}$$

$$\frac{dZ}{dX} = 2 \sin \frac{\pi X^2}{2}$$

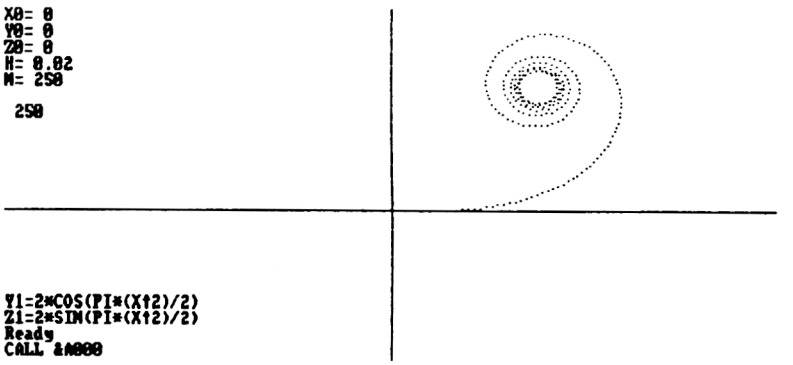
après avoir posé A = 2

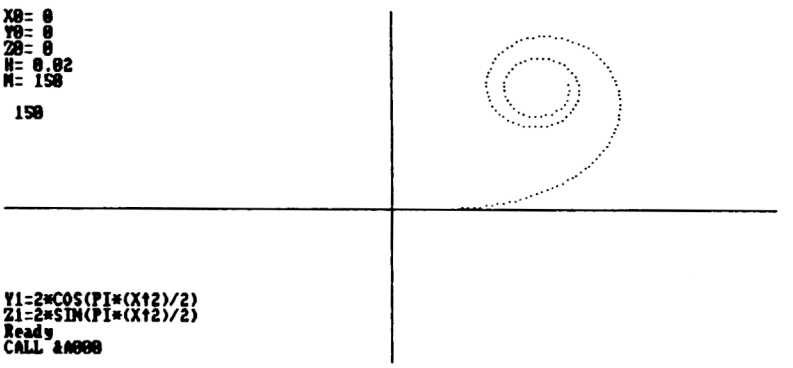
Après RUN, l'ordinateur questionne : X0 = 0 Y0 = 0 Z0 = 0 H = 0.02 M = 150.

Ligne 1000 par Y1 = 2 cos (PI * (Y ↑ 2)/2)
Ligne 1010 par Z1 = 2 sin (PI * (X ↑ 2)/2)
Ligne 2030 : on trace la courbe.

```
10 REM METHODE DE RUNGE-KUTTA
20 REM SPIRALE DE CORNU
25 REM **********************
30 CLS
40 REM CONDITIONS INITIALES
50 REM ********************
60 INPUT "VALEUR DE XO=";XO
70 INPUT "VALEUR DE YO=";YO
80 INPUT "VALEUR DE ZO=";ZO
90 INPUT "VALEUR DE L'INCREMENT H=";H
100 INPUT "NOMBRE D'EVALUATIONS M=";M
110 S=0:DIM A(M):DIM B(M)
115 MODE 2
120 LOCATE 1,1:PRINT "XO=";XO
130 LOCATE 1,2:PRINT "YO=";YO
140 LOCATE 1,3:PRINT "ZO=";ZO
150 LOCATE 1,4:PRINT "H=";H
160 LOCATE 1,5:PRINT "M=";M
170 FOR N=0 TO M
175 LOCATE 1,7:PRINT N
180 X=XO:Y=YO:Z=ZO:GOSUB 1000
190 K1=H*Y1:L1=H*Z1
195 X=XO+H/2:Y=YO+K1/2:Z=ZO+L1/2
200 GOSUB 1000
210 K2=H*Y1:L2=H*Z1
```

```
220 Y=Y0+K2/2:Z=Z0+L2/2
230 GOSUB 1000
240 K3=H*Y1:L3=H*Z1
250 X=X0+H:Y=Y0+K3:Z=Z0+L3
260 GOSUB 1000
270 K4=H*Y1:L4=H*Z1
280 YX=Y0+(K1+2*K2+2*K3+K4)/6
290 ZX=Z0+(L1+2*L2+2*L3+L4)/6
300 A(N)=YX:B(N)=ZX
310 IF ABS(YX)>S THEN S=ABS(YX)
320 IF ABS(ZX)>S THEN S=ABS(ZX)
330 X0=X:Y0=YX:Z0=ZX
340 NEXT N
350 GOTO 1030
500 REM SYST.EQUAT.(LIGNES 1000,1010)
510 REM ****************************
1000 Y1=2*COS(PI*(X^2)/2)
1010 Z1=2*SIN(PI*(X^2)/2)
1015 RETURN
1030 V$="2*COS(PI*(X^2)/2)"
1040 W$="2*SIN(PI*(X^2)/2)"
1050 LOCATE 1,20:PRINT "Y1=";V$
1060 LOCATE 1,21:PRINT "Z1=";W$
2000  PLOT 320,0:DRAWR 0,400
2010  PLOT 0,200:DRAWR 640,0
2020 FOR N=0 TO M
2030 PLOT A(N)/S*190+320,B(N)/S*190+200
2040 NEXT N
```





Le voyage terre-lune selon Jules Vernes

En 1865, le très sérieux «Journal des Débats» publie en feuilleton : «De la Terre à la Lune», puis «Autour de la Lune» de Jules Verne. L'auteur imaginait alors la trajectoire d'un «boulet-wagon» entre les deux astres tiré à partir d'un gigantesque canon situé en Floride.

En 1968, la nuit de Noël, le monde stupéfait suit à la radio et à la télévision les évolutions de la cabine Appollo 8, pilotée par F. Borman, J. Lovell et W. Anders, en orbite circulaire autour de la Lune.

Quelques mois plus tard, le 16 Juillet 1969 à 3 h 56 mn 20 s (heure de Paris), le LEM (baptisé Eagle) de la mission Appollo 11 déposait les astronautes N. Armstrong et E. Aldrin sur le sol lunaire ; leur camarade M. Collins attendant leur retour en orbite lunaire dans la cabine Appollo.

Un siècle plus tard, le rêve fou du Président Barbicane, du capitaine Nicholl et du journaliste Michel Ardan était devenu réalité.

Le problème des trois corps

Dans «Autour de la Lune», le Président Barbicane donne l'équation du mouvement du boulet.

Pour cela, il considère la Terre et la Lune immobiles dans un repère galiléen et applique le théorème de l'énergie cinétique entre les deux astres. L'équation obtenue est évidemment approchée.

Nous avons dans un précédent ouvrage (1) étudié les interactions gravitationnelles et leurs applications à l'étude du mouvement des satellites terrestres.

(1) «Programmes de physique sur AMSTRAD» - Editions Eyrolles.

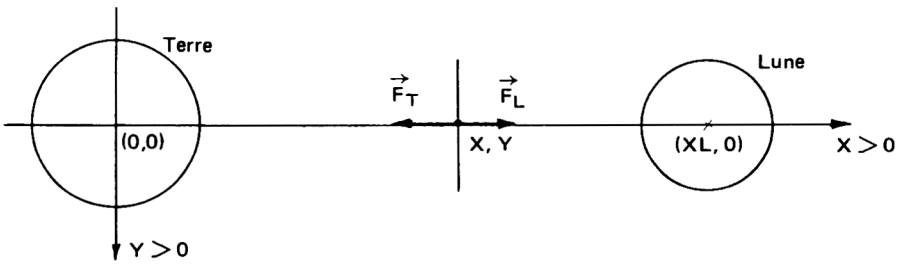
Rappelons que :

— La loi de gravitation exprime que deux masses M1 et M2 situées à une distance d l'une de l'autre, s'attirent mutuellement avec une force d'intensité :

$$f = G \frac{M1M2}{d^2}$$

où G est la constante de gravitation universelle ($G = 6.67 \times 10^{-11}$ SI).

Prenons le centre de la Terre pour origine des axes de coordonnées.



Imaginons un vaisseau spatial de masse M en mouvement entre la Terre et la Lune. Au point de coordonnées X, Y à l'instant t, le vaisseau est soumis aux forces :

— $\overrightarrow{FT}$ due à l'attraction de la Terre,
— $\overrightarrow{FL}$ due à l'attraction de la Lune.

Ecrivons $\vec{F} = \overrightarrow{FT} + \overrightarrow{FL}$

Les composantes de F sont donc :

$$FX = \frac{G.MT.M}{(X^2 + Y^2)^{3/2}} \cdot (-X) + \frac{G.ML.M}{((XL - X)^2 + Y^2)^{3/2}} \cdot (XL - X)$$

$$FY = \frac{G.MT.M}{(X^2 + Y^2)^{3/2}} \cdot (-Y) + \frac{G.ML.M}{((XL - X)^2 + Y^2)^{3/2}} \cdot (-Y)$$

Appliquons la relation fondamentale de la Dynamique $\vec{F} = M\vec{\gamma}$.

Les composantes de l'accélération du vaisseau sont donc :

$$\gamma X = \frac{GMT}{(X^2 + Y^2)^{3/2}} (-X) + \frac{GML}{((XL - X)^2 + Y^2)^{3/2}} (XL - X)$$

$$\gamma Y = \frac{GMT}{(X^2 + Y^2)^{3/2}} (-Y) + \frac{GML}{((XL - X)^2 + Y^2)^{3/2}} \cdot (-Y)$$

Liste des variantes

R1 rayon terrestre au cube
R2 rayon lunaire au cube
MT masse de la Terre
ML masse de la Lune
XL distance Terre-Lune

En ligne 460 la constante K est obtenue par l'opération suivante : 400/XL (400 points sur l'écran correspondent à la distance Terre-Lune XL).

Déroulement du programme

Nous avons montré comment résoudre numériquement les équations différentielles.

Pour calculer la trajectoire, nous utilisons l'algorithme de Runge-Kutta, précédemment exposé (voir la Spirale de Cornu), qui nous permettra une exploitation plus rapide du phénomène.

Ligne (s)	
10 à 260	Titres et introduction des données Le pas de calcul DT sera pris égal à 5 ou 10 mn
300 à 490	On trace la Terre et la Lune (effet de loupe sur les deux astres par rapport à l'échelle de la trajectoire)
500 à 580	Calcul de l'accélération
900 à 1230	Utilisation de l'algorithme
1250 à 2070	On trace la trajectoire

Remarque :

En lignes 530 et 540, le vaisseau percute la Terre ou la Lune.

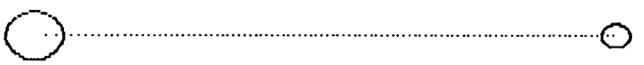
Les calculs effectués en appliquant tout simplement la relation fondamentale de la dynamique sont approximatifs, mais l'ordinateur nous permet de simuler le phénomène très rapidement (environ 20 mn par trajectoire).

```
10 REM VOYAGE TERRE-LUNE
20 REM *****************
30 REM
40 CLS
50 REM SIMULATION DE TRAJECTOIRES
60 REM *************************
70 REM INTRODUCTION DES COORD.DE DEPART
80 INPUT "X=";X:INPUT "Y=";Y
90 REM VITESSE INITIALE
100 INPUT "VX=";VX:INPUT "VY=";VY
110 REM INTODUCTION DES CONSTANTES
```

```
120 G=6.67E-11:MT=5.98E+24:ML=7.35E+22:T=0
130 INPUT "PAS DU CALCUL DT=";DT
140 CLS
150 MODE 2
160 LOCATE 3,17:PRINT "X=";X;"KM"
170 LOCATE 3,18:PRINT "Y=";Y;"KM"
180 LOCATE 3,19:PRINT "VX=";VX;"KM/S"
190 LOCATE 3,20:PRINT "VY=";VY;"KM/S"
200 LOCATE 3,21:PRINT "DT=";DT;"MN"
210 R1=2.586E+20:R2=5.244E+18
220 XL=384000000
230 X=X*1000:Y=Y*1000:VX=VX*1000:VY=VY*1000
240 DT=DT*60
250 DIM A(4):DIM B(4):DIM C(4)
260 DIM D(4):DIM E(4):DIM F(4):DIM N(4)
270 LOCATE 30,2:PRINT"VOYAGE TERRE-LUNE"
300 REM TRACE DE LA TERRE
310 REM *********************
330 FOR I=0 TO 360 STEP 5
340 J=I*PI/180
350 X1=60+20*COS(J):Y1=20*SIN(J)+220
360 PLOT X1,Y1
370 NEXT I
380 REM TRACE DE LA LUNE
390 REM ********************
400 FOR I=0 TO 360 STEP 5
410 J=I*PI/180
420 X1=460+10*COS(J):Y1=10*SIN(J)+220
430 PLOT X1,Y1
450 NEXT I
460 K=400/XL
470 X0=60+K*X:Y0=220-K*Y:PLOT X0,Y0
475 GOSUB 1000
480 GOSUB 2000
490 GOTO 475
500 REM ACCELERATION DU VAISSEAU
510 REM ************************
520 Y1=B(1):Y2=B(2)
530 N1=(Y1*Y1+Y2*Y2)^1.5:IF N1<R1 THEN STOP
540 N2=((ABS(XL-Y1))^2+Y2*Y2)^1.5:IF N2<R2 THEN
OP
550 N(1)=B(3):N(2)=B(4)
560 N(3)=-G*MT*Y1/N1+G*ML*(XL-Y1)/N2
570 N(4)=-G*MT*Y2/N1-G*ML*Y2/N2
580 RETURN
900 REM ALGORITHME DE RUNGE-KUTTA
910 REM *************************
1000 A(1)=X:A(2)=Y:A(3)=VX:A(4)=VY
1010 FOR F=1 TO 4
1020 B(F)=A(F)
1030 NEXT F
1040 GOSUB 520
1060 FOR F=1 TO 4
1070 C(F)=N(F)*DT:B(F)=A(F)+C(F)/2
1080 NEXT F
1090 GOSUB 520
1100 FOR F=1 TO 4
1110 D(F)=DT*N(F):B(F)=A(F)+D(F)/2
1120 NEXT F
1130 GOSUB 520
```

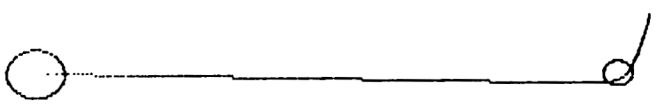
```
1140 FOR F=1 TO 4
1150 E(F)=DT*N(F):B(F)=A(F)+E(F)
1160 NEXT F
1170 GOSUB 520
1180 FOR F=1 TO 4
1190 F(F)=DT*N(F)
1200 A(F)=A(F)+(C(F)+2*(D(F)+E(F))+F(F))/6
1210 NEXT F
1220 X=A(1):Y=A(2):VX=A(3):VY=A(4)
1230 RETURN
1250 REM TRACE DE LA TRAJECTOIRE
1260 REM **********************
2000 X1=60+K*X:Y1=220-K*Y
2010 IF X1>600 THEN GOTO 2050
2020 PLOT X1,Y1
2050 T=T+DT
2060 LOCATE 3,22:PRINT "DUREE-VOYAGE=";T/3600;"HEU
RES"
2070 RETURN
```

VOYAGE TERRE-LUNE



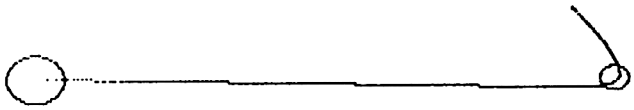
```
X= 8000 KM
Y= 0 KM
VX= 11.5 KM/S
VY= 0 KM/S
DT= 10 MN
DUREE-VOYAGE= 16.6666667 HEURESS
Break in 540
Ready
CALL &A000
```

VOYAGE TERRE-LUNE



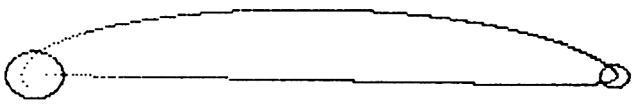
```
X= 8000 KM
Y= 0 KM
VX= 9.94 KM/S
VY= 0.12 KM/S
DT= 10 MN
DUREE-VOYAGE= 75.1666667 HEURESS
Break in 580
Ready
CALL &A000
```

VOYAGE TERRE-LUNE



```
  X= 8000 KM
  Y= 0 KM
  VX= 9.915 KM/S
  VY= 0.12 KM/S
  DT= 10 MN
  DUREE-VOYAGE= 92.6666667 HEURESS
Break in 570
Ready
CALL &A000
```

VOYAGE TERRE-LUNE

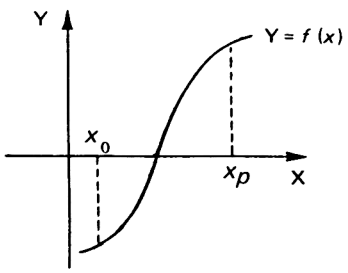


```
  X= 8000 KM
  Y= 0 KM
  VX= 9.904 KM/S
  VY= 0.12 KM/S
  DT= 10 MN
  DUREE-VOYAGE= 165.833333 HEURESS
Break in 530
Ready
CALL &A000
```

Calcul d'une intégrale définie
Méthode de Simpson

Introduction

Etant donnée une fonction $y = f(x)$ et sa représentation graphique.



Soit à calculer l'aire comprise entre la courbe et l'axe des x, limitée aux verticales d'abscisses x_0 et x_p.

On partage l'intervalle $x_p - x_0$ en $N = 2p$ parties.

Soit $E = \dfrac{x_p - x}{N}$. On démontre que l'aire cherchée :

$$S = \int_{x_0}^{x_p} f(x)\,dx$$ est donnée numériquement par :

$$S = \frac{E}{3}\left[(y_0 + y_p) + 4(y_1 + y_3 + \ldots + y_{p-1}) + 2(y_2 + y_4 + \ldots + y_{p-2})\right]$$

Déroulement du programme

Ligne(s)

10 à 70	entrées des données X0, *x*P, N la donnée $y = f(x)$ est introduite en ligne 90. Nous avons pris pour exemple $y = 4/(1 + X \uparrow 2)$
120	calcul de E
150	on fait apparaître les termes figurant dans le crochet
180	c'est le terme E/3 qui est multiplié par le crochet

Exécution : RUN. Entrer X0 = 0 et XP = 1 N = 40. Vous retrouverez le nombre PI.

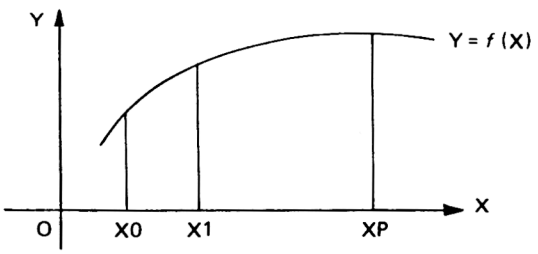
```
10 REM METHODE DE SIMPSON
20 REM ******************
30 REM
40 CLS
50 INPUT "BORNE INFERIEURE X0=";X0
60 INPUT "BORNE SUPERIEURE XP=";XP
70 INPUT "NOMBRE DE DIVISIONS N=";N
80 LOCATE 12,12:PRINT "CALCUL EN COURS"
90 DEF FNY(X)=4/(1+X^2)
100 A$="4/(1+X^2)
110 I=N/2-1
120 E=(XP-X0)/N
130 F1=0:F2=0:X=X0
140 FOR K=1 TO I
150 X=X+E:F1=F1+FNY(X)
160 X=X+E:F2=F2+FNY(X)
170 NEXT K
180 R=E/3*(FNY(X0)+FNY(XP)+4*(F1+FNY(X+E))+2*F2)
185 CLS
190 LOCATE 3,2:PRINT "Y=";A$
200 LOCATE 3,4:PRINT "BORNE INF.X0=";X0
210 LOCATE 3,5:PRINT "BORNE SUP.XP=";XP
220 LOCATE 3,6:PRINT "NOMB.DIV. N=";N
230 LOCATE 3,7:PRINT "REPONSE R=";R
240 STOP
```

```
Y=4/(1+X↑2)

BORNE INF.X0= 0
BORNE SUP.XP= 1
NOMB.DIV. N= 40
REPONSE R= 3.14159266
Break in 240
Ready
CALL &A000
```

```
Y=LOG(X)/X

BORNE INF.X0= 3
BORNE SUP.XP= 5
NOMB.DIV. N= 40
REPONSE R= 0.691670715
Break in 240
Ready
CALL &A000
```

Calcul d'une intégrale définie
Méthode des trapèzes

Rappel du calcul d'une intégrale définie par la méthode des trapèzes



Etant donné une fonction Y = f (X) et sa représentation graphique, soit à calculer l'aire comprise entre la courbe et l'axe des X limitée aux verticales d'abscisses X0 et XP.

On partage l'intervalle en N parties. On démontre que :

$$\int_{X0}^{XP} f(X).dX = \frac{(XP-X0)}{N}\left[\frac{1}{2}f(X0) + \frac{1}{2}f(XP) + f(X1) + \ldots + f(XP-1)\right]$$

Déroulement du programme

Le programme permet de calculer l'intégrale par intervalle choisi.

Ligne(s)

10 à 75	introduction des données
80-85	on définit la fonction
90 à 160	calcul de l'intégrale
165 à 200	l'ordinateur imprime sur l'écran les différents paramètres nécessaires au calcul, ainsi que le résultat de l'intégration

```
10 REM METHODE DES TRAPEZES
20 REM ********************
30 REM
40 CLS
50 INPUT "BORNE INFERIEURE XO=";XO
60 INPUT "BORNE SUPERIEURE XP=";XP
70 INPUT "NOMBRE D'INTERVALLES N=";N
75 LOCATE 12,12:PRINT "CALCUL EN COURS"
80 DEF FNY(X)=4/(1+X^2)
85 A$="4/(1+X^2)"
90 E=(XP-XO)/N
100 X=XO:R=0
110 R=(FNY(XO)+FNY(XP))/2
120 FOR I=1 TO N-1
130 X=X+E
140 R=R+FNY(X)
150 NEXT I
155 CLS
160 R=R*E
161 CLS
165 LOCATE 3,2:PRINT A$
170 LOCATE 3,4:PRINT "BORNE INF.XO=";XO
180 LOCATE 3,5:PRINT "BORNE SUP.XP=";XP
190 LOCATE 3,6:PRINT "NB.D'INTERV. N=";N
200 LOCATE 3,7:PRINT "REPONSE R=";R
```

```
4/(1+X↑2)

BORNE INF.X0= 0
BORNE SUP.XP= 1
NB.D'INTERV. N= 40
REPONSE R= 3.14148849
Ready
CALL &A000
```

```
LOG(X)/X

BORNE INF.X0= 3
BORNE SUP.XP= 5
NB.D'INTERV. N= 40
REPONSE R= 0.691667922
Ready
CALL &A000
```

Recherche des zéros d'un polynôme
Méthode de Bairstow

Soit $p(x) = a_n x^n + a_{n-1} x^{n-1} + \ldots\ldots + a_0, a_n \neq 0$

un polynôme de degré n dont les coefficients $a_n, a_{n-1}, \ldots a_0$ sont réels.

Avec cette restriction, les racines de ce polynôme sont, soit réelles, soit complexes conjuguées. On peut donc l'écrire sous la forme :

$$p(x) = a_n (x^2 + p_1 x + q_1)^{m_1} . (x^2 + p_2 x + q_2)^{m_2} \ldots . (x - x_r)^r \ldots . (x - x_i)^i$$

$$2(m_1 + m_2 + \ldots) + (r + \ldots + i) = n$$

Mettons ce polynôme sous la forme :

$$p(x) = (x_2 + px + q)\, p_1(x) + R(p, q)\, x + S(p, q).$$

Ceci est toujours possible puisque l'on a :

(en posant : $p_1(x) = b_n x^{n-2} + b_{n-1} x^{n-3} + \ldots\ldots + b_2$)

$b_n = a_n$

$b_{n-1} = a_{n-1} - pb_n$

.

.

.

$b_k = a_k - pb_{k+1} - qb_{k+2}$ avec $k = n - 2, \ldots, 0$

En supposant b_0 et b_1 définis de la même manière :

$R(p, q) = b_1 = a_1 - pb_2 - qb_3$

$S(p, q) = b_0 + pb_1 = a_0 - qb_2 = (a_0 - pb_1 - qb_2) + pb_1$

En définissant des formules de récurrence analogues :

$C_n = b_n$

$C_{n-1} = b_{n-1} - pc_n$

.

.

.

.

$C_2 = b_2 - pc_3 - qc_4$

$C_1 = -pc_2 - qc_3$

On démontre dans ces conditions :

$$R_p = \frac{\partial R}{\partial p} = -c_2 \quad R_q = \frac{\partial R}{\partial q} = -c_3 \quad S_p = \frac{\partial S}{\partial p} = c_1 - pc_2 \quad S_q = \frac{\partial S}{\partial q} = -c_2 - pc_3$$

La méthode de résolution est alors la suivante :

— par une méthode itérative, chercher les valeurs de p et de q qui annulent R (p, q) et S (p, q). p et q seront ainsi les coefficients du trinôme que l'on peut mettre en facteur et dont les racines sont aussi racines de $p(x)$.

Cette méthode itérative est déduite de celle de Newton et elle s'applique ici à un système de 2 équations à 2 variables*.

On démontre que, après un certain nombre d'itérations, p_n et q_n tendent vers les valeurs de p et q cherchées :

$$p_n = p_{n-1} - \frac{R}{R_p S_q - S_p R_q} S_q R(p_{n-1}, q_{n-1}) - R_q S(p_{n-1}, q_{n-1})$$

$$q_n = q_{n-1} - \frac{1}{R_p S_q - S_p R_q} - S_p R(p_{n-1}, q_{n-1}) + R_p S(p_{n-1}, q_{n-1})$$

— Connaissant ainsi les valeurs de p et de q, il est possible, en divisant le polynôme initial par ce trinôme, d'en déduire un polynôme de degré plus faible.

— L'opération est réitérée jusqu'à l'obtention d'un polynôme de degré 1 ou 2.

Il est certain que la précision se dégrade au cours du calcul.

* Il faut que R (p, q) et S (p, q) s'annulent.

En développant en série :

$$R(p, q) = R(p_0, q_0) + (p - p_0)\frac{\partial R}{\partial p}(p_0, q_0) + (q - q_0)\frac{\partial R}{\partial q}(p_0, q_0) + \ldots$$

$$S(p, q) = S(p_0, q_0) + (p - p_0)\frac{\partial S}{\partial p}(p_0, q_0) + (q - q_0)\frac{\partial S}{\partial q}(p_0, q_0) + \ldots$$

Ce qui peut s'écrire :

$$\begin{bmatrix} R(p, q) \\ S(p, q) \end{bmatrix} = \begin{bmatrix} R(p_0, q_0) \\ S(p_0, q_0) \end{bmatrix} \begin{bmatrix} \frac{\partial R}{\partial p}(p_0, q_0) & \frac{\partial R}{\partial q}(p_0, q_0) \\ \frac{\partial S}{\partial p}(p_0, q_0) & \frac{\partial S}{\partial q}(p_0, q_0) \end{bmatrix} \begin{bmatrix} p - p_0 \\ q - q_0 \end{bmatrix} + \ldots$$

En appliquant la méthode de Newton à ce système à 2 dimensions, on pose :

$$R(p, q) = 0$$
$$S(p, q) = 0$$

D'autre part, avec :

$$\frac{\partial R}{\partial p}(p_0, q_0) = R_p \qquad \frac{\partial R}{\partial p}(p_0, q_0) = R_q$$

$$\frac{\partial S}{\partial p}(p_0, q_0) = S_p \qquad \frac{\partial S}{\partial q}(p_0, q_0) = S_q$$

Pour simplifier l'écriture, on a :

$$-\begin{bmatrix} R(p_0, q_0) \\ S(p_0, q_0) \end{bmatrix} = \begin{bmatrix} R_p & R_q \\ S_p & S_q \end{bmatrix} \begin{bmatrix} p - p_0 \\ q - q_0 \end{bmatrix}$$

D'où, après inversion de la matrice :

$$\begin{bmatrix} p \\ q \end{bmatrix} = \begin{bmatrix} p_0 \\ q_0 \end{bmatrix} - \begin{bmatrix} R_p & R_q \\ S_p & S_q \end{bmatrix}^{-1} \cdot \begin{bmatrix} R \\ S \end{bmatrix}_{(p_0, q_0)}$$

On sait que :

$$\begin{bmatrix} R_p & R_q \\ S_p & S_q \end{bmatrix}^{-1} = \frac{\begin{bmatrix} S_p & -R_q \\ -S_p & R_p \end{bmatrix}}{R_p\ S_q - R_q\ S_p}$$

Partant du couple p_{n-1}, q_{n-1}, on trouve une nouvelle approche :

$$\begin{bmatrix} p_n \\ q_n \end{bmatrix} = \begin{bmatrix} p_{n-1} \\ q_{n-1} \end{bmatrix} - \begin{bmatrix} R_p & R_q \\ S_p & S_q \end{bmatrix}^{-1} \cdot \begin{bmatrix} R \\ S \end{bmatrix}_{p_{n-1},\ q_{n-1}}$$

Rappelons que la méthode de Newton conduit à écrire :

$$f(x) = f(x_0) + (x - x_0) \cdot f'(x_0)$$

Puis, les racines de $f(x) = 0$ amènent à écrire :

$$0 = f(x_0) + (x - x_0) \cdot f'(x_0)$$

D'où $x = x_0 - \dfrac{f(x_0)}{f'(x_0)}$

Soit l'algorithme général :

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)}$$

qui converge vers une racine réelle de $f(x)$, si elle existe.

Programme

```
10 REM ************************
20 REM *                      *
30 REM * racines d'UN POLYNOME *
40 REM *                      *
50 REM ************************
60 REM d'apres W.Luther
70 REM Algorithmes
80 REM editions Sybex
90 REM *** ps:precision de solution du systeme r(p
,q)=0,s(p,q)=0***
100 REM *** pn:precision de la methode de newton(l
inearisation)***
110 ps =0.0001:pn=0.000000001:ni=70:nj=ni
120 REM *** n degre du polynome p(x)=a(n)x^n+a(n-1
)x^(n-1)+...+a(0)***
130 REM *** p(x)=(x^2+px+q)*p1(x)+r(p,q)x+s(p,q) *
**
140 INPUT " degre de polynome n=";n
150 PRINT #8," degre de polynome n=";n
160 REM ***introduction des coeficients***
170 DIM a(n),b(n),c(n)
180 PRINT"coeficients a(n),a(n-1),....a(0):"
190 FOR k=n TO 0 STEP-1
200 READ a(k):PRINT" ";a(k),
210 PRINT #8,"";k,a(k)
220 NEXT
230 REM *** introduction des valeurs initiales p,q
 du trinome  x^2+px+q ***
240 PRINT
250 INPUT" p=";p
260 INPUT" q=";q
270 PRINT:PRINT "racines du polynome"
280 PRINT #8:PRINT #8,"racines du polynome"
290 PRINT "methode de bairstow"
300 PRINT #8,"methode de bairstow"
310 IF n>2 THEN 510
320 IF n=1 THEN PRINT-a(0)/a(1)
```

```
330 IF n=1 THEN PRINT #8,-a(0)/a(1):END
340 p1=a(1)/a(2):q1=a(0)/a(2)
350 re =-p1/2:im=re*re-q1:IF im<0 THEN 410
360 PRINT re+SQR(im)
370 PRINT #8,re+SQR(im)
380 PRINT re-SQR(im)
390 PRINT #8,re-SQR(im)
400 GOTO 450
410 im=SQR(-im)
420 PRINT re;"+j*";im
430 PRINT #8,re;"+j*";im
440 PRINT #8,re;"-j*";im
450 n=n-2:IF n=0 THEN END
460 FOR k =0 TO n
470 a(k)=b(k+2)
480 NEXT
490 GOTO 310
500 REM ***methode de bairstow ***
510 ni=0
520 b(n)=a(n):c(n)=a(n)
530 b(n-1)=a(n-1)-p*b(n)
540 c(n-1)=b(n-1)-p*c(n)
550 FOR k=n-2 TO 0 STEP -1
560 b(k)=a(k) -p*b(k+1) -q*b(k+2)
570 c(k)=b(k) -p*c(k+1) -q*c(k+2)
580 NEXT
590 c(1)=c(1)-b(1)
600 REM *** iteration ***
610 d=c(2)*c(2)-c(1)*c(3)
620 p1=p+(c(2)*b(1)-c(3)*b(0))/d
630 q1=q+(c(2)*b(0)-c(1)*b(1))/d
640 IF (ABS(b(1))<ps)AND(ABS(b(0)+p*b(1)) <ps) THE
N 350
650 IF (ABS(p1-p)<pn)AND(ABS(q1-q)<pn)THEN 680
660 ni=ni+1:IF ni>nj THEN 700
670 p=p1:q=q1:GOTO 530
680 PRINT"precision";ps;"non obtenue, choisissez d
'autres valeurs initiales"
690 END
700 PRINT"pas de convergences apres";nj;"iteration
s,augmentez ni.":END
710 REM *** a(n), a(n-1),..., a(0)***
720 DATA 1,0,13,0,26,0,-1
```

```
 degre de polynome n= 8
 8              1
 7              0
 6              0
 5              0
 4              0
 3              0
 2              0
 1              0
 0             -1

racines du polynome
methode de bairstow
 1
-1
-1.20914E-07 +j* 1.00000012
-1.20914E-07 -j* 1.00000012
 0.707106875 +j* 0.707106988
 0.707106875 -j* 0.707106988
-0.707106887 +j* 0.707106785
-0.707106887 -j* 0.707106785
```

```
 degre de polynome n= 5
 5              324
 4             -351
 3              108
 2              6
 1             -8
 0              1

racines du polynome
methode de bairstow
 0.33816458
 0.327235611
 0.334597249 +j* 7.77077E-03
 0.334597249 -j* 7.77077E-03
-0.249999126

p=1,q=1
```

```
 degre de polynome n= 5
 5              324
 4             -351
 3              108
 2              6
 1             -8
 0              1

racines du polynome
methode de bairstow
 0.333227791 +j* 7.94569E-03
 0.333227791 -j* 7.94569E-03
 0.344776058
 0.322312802
-0.250000313
p=-0.66 , q=0.111
les racines exactes sont :
1/3,1/3,1/3,1/3,-1/4
```

```
 degre de polynome n= 6
 6              1
 5              11
 4              45
 3              84
 2              70
 1              21
 0              1

racines du polynome
methode de bairstow
-0.502978504
-1.29079022
-5.81176E-02
-2.24108785
-3.13613972
-3.77090324
```

```
degre de polynome n= 6
6              1
5              0
4              13
3              0
2              26
1              0
0             -1

racines du polynome
methode de bairstow
 0.194286
-0.194286
-5.26951E-10 +j* 1.58695514
-5.26951E-10 -j* 1.58695514
 1.27590E-10 +j* 3.24335019
 1.27590E-10 -j* 3.24335019
```

Statistiques et calcul de probabilité Loi de Gauss

Introduction

Le statisticien doit d'abord définir avec précision l'ensemble de référence sur lequel vont porter les observations. Cet ensemble s'appelle : « L'ensemble statistique ou population ».

Pour le physicien, le terme « population » désigne un ensemble de faits expérimentaux susceptibles d'être reproduits dans des conditions absolument semblables.

Tous les facteurs possibles de variation qui n'entrent pas dans la définition de la « population », c'est-à-dire non contrôlés pendant l'expérimentation doivent être attribués au hasard.

En fait, le calcul statistique ne s'applique, la plupart du temps, qu'à des populations dites « normales » représentées par la loi de Gauss.

Il s'agit d'une loi de probabilité d'une variable aléatoire x continue, variable qui est définie de $-\infty$ à $+\infty$.

Sa densité de probabilité est de la forme :

$$f(X) = \frac{1}{\sqrt{2\pi\sigma^2}} \cdot \exp. \frac{(-X - Xmoy)^2}{2\sigma^2}$$

Cette courbe est caractérisée par :

— La moyenne arithmétique Xmoy des n résultats individuels X.

— La variance :

$$v = \frac{(X - Xmoy)^2}{n} = \overline{\Delta X^2}$$

avec $X = X - Xmoy$.

— L'écart type $\sigma = \sqrt{v}$ $(E = \sigma)$

On peut définir aussi :

— Le coefficient de variation $k = \frac{100}{X\,moy}$

— L'écart moyen ou erreur type : $Emoy = \frac{E}{\sqrt{n}}$

— L'écart type affecte un résultat isolé et caractérise la précision de ce même résultat isolé.

— L'écart moyen affecte une moyenne et caractérise la précision de la méthode analytique utilisée.

En Physique :

— Le terme «précision» désigne le degré de reproductibilité fonction seulement des erreurs accidentelles.

— L'erreur systématique caractérise l'exactitude ou degré de justesse de la méthode.

Les erreurs systématiques sont indépendantes du hasard et se produisent toujours dans le même sens.

Les erreurs accidentelles, au contraire, ont lieu dans les deux sens avec une probabilité égale.

Loi de Gauss généralisée

Une variable aléatoire continue X suit une loi de Gauss généralisée si elle a pour densité de probabilité :

$$f(X) = \frac{1}{\sqrt{2\pi\sigma^2}} \cdot \exp \frac{-(X - Xmoy)}{2\sigma^2}$$

La représentation graphique de $f(X)$ est symétrique par rapport à $(X - Xmoy)$.

Loi de Gauss centrée réduite

On passe de :

$$f(x) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp \frac{-(X - Xmoy)}{2\sigma^2} \quad \text{à} \quad f(t) = \frac{1}{\sqrt{2\pi}} \exp(-t^2/2)$$

par le changement de variable : $t = \frac{X - Xmoy}{\sigma}$

Remarque :

A l'occasion de ce changement de variable, le terme :

$\exp -\left(\frac{X - moy}{2\sigma^2}\right)$ devient $\exp(-t^2/2)$ et le terme constant :

$\frac{1}{\sqrt{2\pi\sigma^2}}$ tiré de $f(X)$ est multiplié par σ.

En effet, la division par σ lors du changement de variable oblige pour conserver les surfaces et en conséquence les probabilités, à multiplier les ordonnées par σ.

Les problèmes qui concernent la loi de Gauss généralisée peuvent donc être résolus à l'aide de la loi de Gauss centrée réduite.

– Lorsque t croît en valeur absolue, le terme $\exp(-t^2/2)$ devient rapidement asymptotique à l'axe des abscisses.

– La fonction $f(t)$ passe un maximum pour $t = 0$ maximum égal à :

$$f(0) = \frac{1}{\sqrt{2\pi}} \exp(0) = \frac{1}{\sqrt{2\pi}} = 0{,}40$$

– La fonction $f(t)$ passe par deux points d'inflexion d'abscisses respectives -1 et $+1$.

Déroulement du programme 1

Ligne(s)

10 à 70 entrées des constantes :

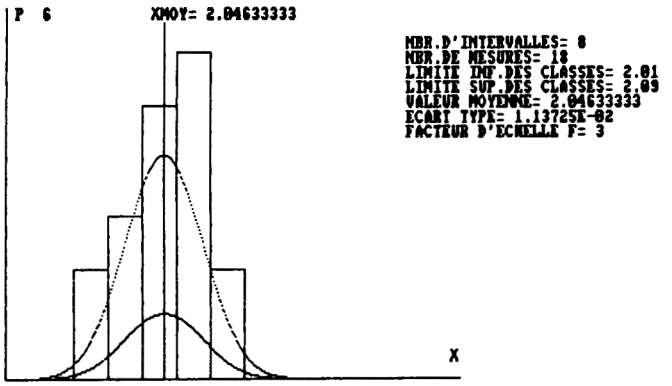
. nombre d'intervalles : N
. nombre de mesures : M
. pas du programme : P
. limite inférieure des classes : K
. facteur d'échelle (dans le sens vertical) : F

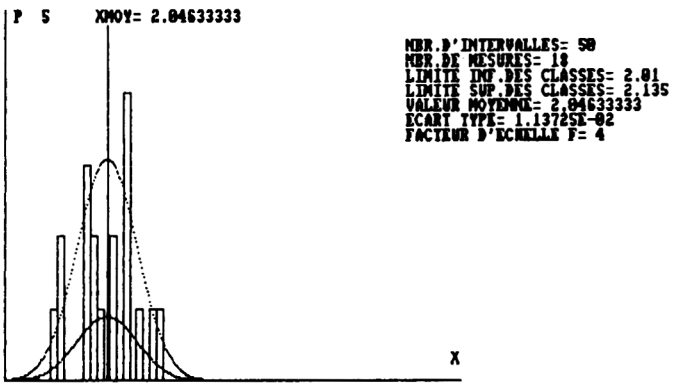
120-130-140 entrées des données numériques concernant la série étudiée. Nous avons rentré une série de mesures calorimétriques destinées à déterminer une constante d'étalonnage

150 à 450 l'ordinateur trace automatiquement l'histogramme de la série étudiée et indique la valeur moyenne

500 à 560 on fait apparaître les différents paramètres nécessaires au bon déroulement du programme

```
10 REM HISTOGRAMME D'UNE SERIE DE MESURES
20 REM ***********************************
25 CLS:MODE 2
30 INPUT "NOMBRE DE MESURES M=";M
40 INPUT "NOMBRE D'INTERVALLES N=";N
50 INPUT "PAS DU CALCUL P=";P
60 INPUT "LIMITE INFERIEURE DES CLASSES K=";K
70 INPUT "FACTEUR D'ECHELLE F=";F
80 I=0:CLS
85 LI=K
90 FOR D=1 TO N
95 G=0:X=0:H=0:V=0
100 FOR X=1 TO M
110 READ Y
120 DATA 2.043,2.042,2.053,2.067,2.029,2.065
130 DATA 2.059,2.048,2.054,2.039,2.050,2.055
140 DATA 2.030,2.042,2.027,2.053,2.040,2.038
150 H=Y+H
160 IF Y>=K AND Y<=K+P THEN G=G+1
170 NEXT X
180 I=I+G
190 C1=G*350/(M/F)
200 IF C1>350 THEN GOTO 1000
230 PLOT 0,0:DRAWR 400,0
240 PLOT 0,0:DRAWR 0,400
245 LOCATE 2,1:PRINT "P":LOCATE 50,24:PRINT "X"
250 PLOT D*(300/(N+2)),0:DRAWR 0,G*350/(M/F)
260 DRAWR 300/(N+2),0:DRAWR 0,-(G*350/(M/F))
270 K=K+P:RESTORE 110
280 NEXT D
290 LOCATE 4,1:PRINT CINT(M/F)
300 XMOY=H/M
310 FOR I=1 TO M
320 READ Y
330 V=((Y-H/M)*(Y-H/M))+V
340 NEXT I
350 J=(((N+1)*P-(K-XMOY))*300/(N+2))/P
360 PLOT J,0:DRAWR 0,400
370 LOCATE CINT(J/8) ,1:PRINT "XMOY=";XMOY
380 E=SQR(V/M)
390 FOR I=CINT(300/(N+2)) TO 300
400 X=P*(((N+2)/300)*I-1)+LI
410 T=(X-XMOY)/E
420 G1=(1/SQR(2*PI))*EXP(-T*T/2)
430 G2=(1/E)*(1/SQR(2*PI))*EXP(-(X-XMOY)*(X-XMOY)/
(2*E^2))
440 PLOT I,G1*600:PLOT I,2*G2
450 NEXT I
500 LOCATE 45,3:PRINT "NBR.D'INTERVALLES=";N
510 LOCATE 45,4:PRINT "NBR.DE MESURES=";M
520 LOCATE 45,5:PRINT "LIMITE INF.DES CLASSES=";LI
530 LOCATE 45,6:PRINT "LIMITE SUP.DES CLASSES=";LI
+N*P
540 LOCATE 45,7:PRINT "VALEUR MOYENNE=";XMOY
550 LOCATE 45,8:PRINT "ECART TYPE=";E
555 LOCATE 45,9:PRINT "FACTEUR D'ECHELLE F=";F
560 STOP
1000 LOCATE 3,3:PRINT "F TROP GRAND"
1010 RESTORE 110
1020 FOR W=1 TO 1000:NEXT W
1030 GOTO 60
```



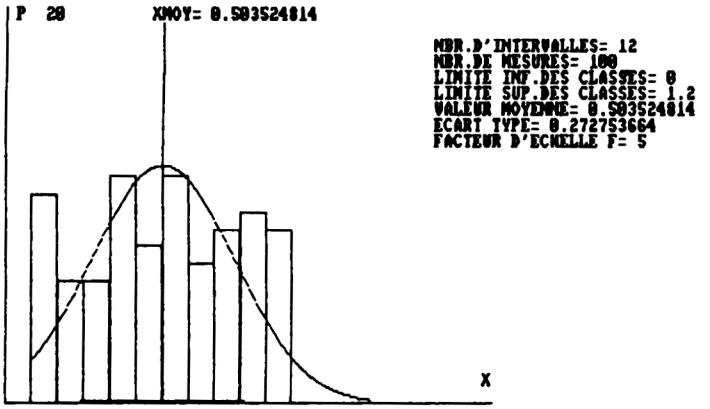


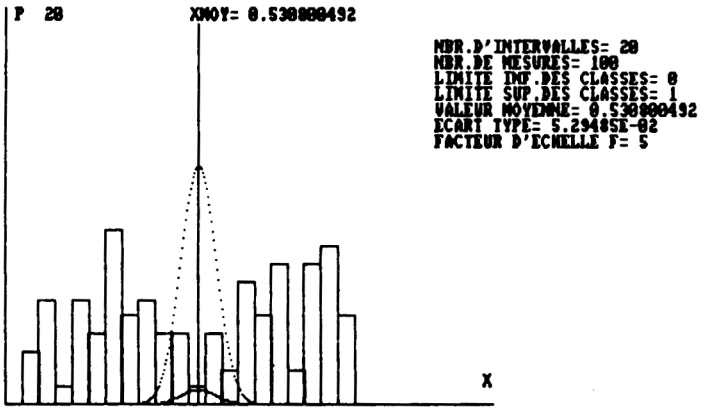
Programme 2

Le programme 2 permet d'étudier la fonction RND de l'AMSTRAD.

```
5 REM ETUDE DE LA FONCTION RND
6 REM ************************
7 CLS:LOCATE 1,1:PRINT "APPUYER SUR A"
8 A=RND
9 A$=INKEY$
10 IF A$<>"A" THEN GOTO 8
25 CLS:MODE 2
30 INPUT "NOMBRE DE MESURES M=";M:DIM A(M)
40 INPUT "NOMBRE D'INTERVALLES N=";N
50 INPUT "PAS DU CALCUL P=";P
60 INPUT "LIMITE INFERIEURE DES CLASSES K=";K
70 INPUT "FACTEUR D'ECHELLE F=";F
```

```
75 LI=K
80 I=0:CLS
85 FOR X=1 TO M:A(X)=RND
86 NEXT X
90 FOR D=1 TO N
95 G=0:X=0:H=0:V=0
100 FOR X=1 TO M
110 Y=A(X)
150 H=Y+H
160 IF Y>=K AND Y<=K+P THEN G=G+1
170 NEXT X
180 I=I+G
190 C1=G*350/(M/F)
200 IF C1>350 THEN GOTO 1000
230 PLOT 0,0:DRAWR 400,0
240 PLOT 0,0:DRAWR 0,400
245 LOCATE 2,1:PRINT "P":LOCATE 50,24:PRINT "X"
250 PLOT D*(300/(N+2)),0:DRAWR 0,G*350/(M/F)
260 DRAWR 300/(N+2),0:DRAWR 0,-(G*350/(M/F))
270 K=K+P:RESTORE 110
280 NEXT D
285 IF I<M THEN GOTO 2000
290 LOCATE 4,1:PRINT CINT(M/F)
300 XMOY=H/M
310 FOR I=1 TO M
320 Y=A(F)
330 V=((Y-H/M)*(Y-H/M))+V
340 NEXT I
350 J=(((N+1)*P-(K-XMOY))*300/(N+2))/P
360 PLOT J,0:DRAWR 0,400
370 LOCATE CINT(J/8) ,1:PRINT "XMOY=";XMOY
380 E=SQR(V/M)
390  FOR I=CINT(300/(N+2)) TO 300
400 X=P*(((N+2)/300)*I-1)+LI
410 T=(X-XMOY)/E
420 G1=(1/SQR(2*PI))*EXP(-T*T/2)
430 G2=(1/E)*(1/SQR(2*PI))*EXP(-(X-XMOY)*(X-XMOY)/
(2*E^2))
440 PLOT I,G1*600:PLOT I,2*G2
450 NEXT I
500 LOCATE 45,3:PRINT "NBR.D'INTERVALLES=";N
510 LOCATE 45,4:PRINT "NBR.DE MESURES=";M
520 LOCATE 45,5:PRINT "LIMITE INF.DES CLASSES=";LI
530 LOCATE 45,6:PRINT "LIMITE SUP.DES CLASSES=";LI
+N*P
540 LOCATE 45,7:PRINT "VALEUR MOYENNE=";XMOY
550 LOCATE 45,8:PRINT "ECART TYPE=";E
555 LOCATE 45,9:PRINT "FACTEUR D'ECHELLE F=";F
560 STOP
1000 LOCATE 3,3:PRINT "F TROP GRAND"
1010 RESTORE 110
1020 FOR W=1 TO 1000:NEXT W
1030 GOTO 60
2000 LOCATE 3,3:PRINT "N OU P MAL CHOISI"
2010 FOR W=1 TO 1000 :NEXT W
2020 GOTO 40
```





Programme 3

Le programme 3 permet de calculer la population par intervalle choisi par application de la loi de Gauss centrée réduite (on utilise la méthode des trapèzes).

```
10 REM LOI DE GAUSS
20 REM ************
30 REM
40 REM CACUL DE LA POPULATION
50 REM PAR INTERVALLE CHOISI
60 CLS:MODE 2
70 INPUT "NOMBRE DE MESURES M=";M
80 INPUT "VALEUR MOYENNE XMOY=";XMOY
90 INPUT "ECART TYPE E=";E
```

```
100 INPUT "NOMBRE D'INTERVALLES (INTEGRATION) N=";
N
110 INPUT "VALEUR INF.DE L'INTERVALLE XO=";XO
120 INPUT "VALEUR SUP.DE L'INTERVALLE XP=";XP
130 DEF FNY(X)=(1/E)*(1/SQR(2*PI))*EXP(-(X-XMOY)*(
X-XMOY)/(2*E^2))
140 R=(XP-XO)/N
150 X=XO:S=0
160 S=(FNY(XO)+FNY(XP))/2
170 FOR I=1 TO N-1
180 X=X+R
190 S=S+FNY(X)
200 NEXT I
210 S=S*R
220 PRINT
230 PRINT "VALEUR DE L'INTEGRALLE S=";S
240 PRINT "POPULATION CORRESPONDANTE P=";S*M
250 PRINT "POPULATION ARRONDIE PA=";CINT(S*M)
260 STOP
270 REM ETUDE D'UN NOUVEL INTERVALLE:FAIRE GOTO 11
0
```

```
NOMBRE DE MESURES M=? 18
VALEUR MOYENNE XMOY=? 2.0463
ECART TYPE E=? .0113
NOMBRE D'INTERVALLES (INTEGRATION) N=? 20
VALEUR INF.DE L'INTERVALLE X0=? 2.01
VALEUR SUP.DE L'INTERVALLE XP=? 2.03

VALEUR DE L'INTEGRALLE S= 7.40536E-02
POPULATIOM CORRESPONDANTE P= 1.33296422
POPULATIOM ARRONDIE PA= 1
Break in 260
Ready
GOTO 110
VALEUR INF.DE L'INTERVALLE X0=? 2.03
VALEUR SUP.DE L'INTERVALLE XP=? 2.04

VALEUR DE L'INTEGRALLE S= 0.213999484
POPULATIOM CORRESPONDANTE P= 3.85199071
POPULATIOM ARRONDIE PA= 4
Break in 260
Ready
CALL &A000
```

```
Ready
GOTO 110
VALEUR INF.DE L'INTERVALLE X0=? 2.04
VALEUR SUP.DE L'INTERVALLE XP=? 2.05

VALEUR DE L'INTEGRALLE S= 0.339693382
POPULATIOM CORRESPONDANTE P= 6.11448088
POPULATIOM ARRONDIE PA= 6
Break in 260
Ready
GOTO 110
VALEUR INF.DE L'INTERVALLE X0=? 2.01
VALEUR SUP.DE L'INTERVALLE XP=? 2.08

VALEUR DE L'INTEGRALLE S= 0.997742948
POPULATIOM CORRESPONDANTE P= 17.9593731
POPULATIOM ARRONDIE PA= 18
Break in 260
Ready
CALL &A000
```

Système d'équations linéaires

On cherche la solution $(x_1 \ldots\ldots x_n)$ du système :

$$a_{11}\, x_1 + a_{12}\, x_2 + \ldots\ldots\ldots + a_{1n}\, x_n = b_1$$
$$a_{21}\, x_1 + a_{22}\, x_2 + \ldots\ldots\ldots + a_{2n}\, x_n = b_2$$
$$\vdots$$
$$a_{n1}\, x_1 + a_{n2}\, x_2 + \ldots\ldots\ldots + a_{nn}\, x_n = b_n$$

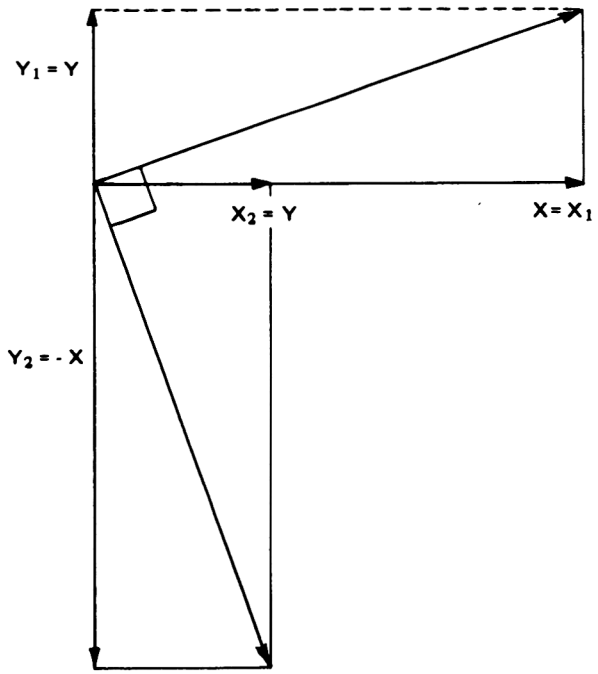
qui peut s'écrire avec la notation matricielle :

$$\begin{bmatrix} a_{11}\, a_{12} \ldots\ldots\ldots a_{1n} \\ a_{21}\, a_{22} \ldots\ldots\ldots a_{2n} \\ \vdots \\ a_{n1}\, a_{n2} \ldots\ldots\ldots a_{nn} \end{bmatrix} \times \begin{bmatrix} x_1 \\ x_2 \\ \\ x_n \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ \\ b_n \end{bmatrix}$$

soit $A.x = b$.

A est une matrice carrée $n \times n$, x et b 2 vecteurs à n composantes.

Il est facile de voir (figure ci-dessous) que, quand deux vecteurs sont orthogonaux, alors leur produit scalaire est nul :
$X_1 X_2 + Y_1 Y_2 = 0 \quad (= XY - YX)$.



Ceci se généralisant pour n dimensions, la relation (1) peut s'écrire :

$$-1.b_1 + a_{11}\,x_1 + a_{12}\,x_2 + \ldots\ldots\ldots + a_{1n}\,x_n = 0$$

$$\vdots$$

$$-1.b_n + a_{n1}\,x_1 + a_{n2}\,x_2 + \ldots\ldots\ldots + a_{nn}\,x_n = 0$$

Ce qui revient à dire que le vecteur $(-1, x_1, x_2, \ldots, x_n)$ doit être orthogonal à tous les autres :
$(b_1, a_{11}, a_{12}, \ldots, a_{1n}) \ldots\ldots (b_n, a_{n1}, \ldots\ldots a_{nn})$.

On considère alors la matrice $(n+1) \,.\, (n+1)$:

$$\begin{bmatrix} 0, b_1, b_2, \ldots\ldots, b_n \\ b_1, a_{11}, a_{12}, \ldots\ldots, a_{1n} \\ \vdots \\ b_n, a_{n1}, a_{n2}, \ldots\ldots, a_{nn} \end{bmatrix} \quad \begin{bmatrix} A_{(0)} \\ A_{(1)} \\ \\ A_{(n)} \end{bmatrix}$$

et l'on utilise l'algorithme de Schmidt qui permet de rendre la première ligne A (0) orthogonale à toutes les autres. Dans ce cas évidemment, la première ligne sera devenue :

$$[-1\ x_1\ x_2 \ldots\ldots x_n] \qquad \text{(solution recherchée)}$$

sous la forme :

$$[a_{00}\ a_{01} \ldots\ldots a_{0n}]$$

et donc :

$$x_1 = -\frac{a_{01}}{a_{00}}\ ,\ x_2 = -\frac{a_{02}}{a_{00}}\ ,\ \ldots\ldots\ ,\ x_n = -\frac{a_{0n}}{a_{00}}$$

Ce procédé consiste à normaliser la dernière ligne A (n) :

$$\frac{A\ (n)}{|A\ (n)|} \rightarrow A\ (n)$$

puis à remplacer A (n — 1) :

$$A\ (n-1) - [A\ (n).A\ (n-1)] \cdot A\ (n) \rightarrow A\ (n-1) \rightarrow \frac{A\ (n-1)}{|A(n-1)|}$$

etc.

Cet algorithme est répété jusqu'à ce que A (0) soit suffisamment orthogonal à tous les autres vecteurs.

On détermine ensuite les résidus :

$$r = b - A.x$$

Posant $A.x_1 = r$ (solution calculée de façon identique), il est certain que $x_1 + x$ est meilleure solution que x_1, puisque :

$$A.x_1 = b - Ax \qquad A\ (x + x_1) = b$$

Le programme

Les données sont introduites dans le sous-programme en lignes 610-700. Pour plus de commodité, elles se trouvent dans les DATA des lignes 1020 à 1100.

Le programme proprement dit commence en ligne 720 (II = nombre d'itérations). Une orthonormalisation est ensuite effectuée (sous-programme 70-510) et l'erreur conséquente estimée.

Une première solution est alors estimée (800-830), un calcul des résidus est fait (840-950), leur valeur est affichée (930) et éventuellement réutilisée pour une nouvelle boucle (ligne 110).

La précision P peut être choisie égale à 1E — 9.

```
10 INPUT "ordre du systeme :";N
20 INPUT "precision :";P
30 DIM z(N,N),a(N,N),b(N),x(N),sm(N)
40 GOSUB 530
50 GOSUB 720
60 REM orthonormalisation
70 FOR k=0 TO N
80 FOR i= 0 TO N
90 a(k,i)=z(k,i)
100 NEXT
110 a(k,0)=sm(k):a(0,k)=sm(k)
120 NEXT
130 ni=0
140 FOR g=n TO 0 STEP -1
150 kz=0:m=0
160 FOR r=n TO g STEP -1
170 ps=kz:kz=0:l=r+m:m=1
180 FOR i=0 TO n
190 ss=a(g,i)-a(l,i)*ps
200 a(g,i)=ss:kz=ss*a(r,i)+kz
210 NEXT i
220 NEXT r
230 IF kz=0 THEN PRINT"det a=0 ?":END
240 ps=1/SQR(kz)
250 FOR i = 0 TO n
260 a(g,i)=a(g,i)*ps
270 NEXT i
280 NEXT g
290 ps=0
300 FOR g=1 TO n
310 kz=0
320 FOR i =0 TO n
330 kz=a(0,i)*a(g,i)+kz
340 NEXT
350 ps=ABS(kz)+ps
360 NEXT
370 kz=ps/n
380 PRINT"orthogonalite solution=";kz
390 ni=ni+1
400 IF (kz>p) AND (ni<4) THEN 140
410 IF kz < p THEN 430
420 PRINT"precision ";p;"non obtenue"
430 IF ABS(a(0,0)) > 1E-10 THEN 450
440 PRINT"systeme incompatible ?":PRINT
450 IF a(0,0)=0 THEN END
460 s=-1/a(0,0)
470 REM solution
480 FOR k=1 TO n
```

```
490 a(0,k)=a(0,k)*s
500 NEXT
510 RETURN
520 REM initialisation
530 FOR k=1 TO n
540 FOR i=1 TO n
550 READ z(k,i)
560 NEXT
570 NEXT
580 FOR k=1 TO n
590 READ b(k)
600 NEXT
610 PRINT"systeme d'equations lineaires"
620 FOR k=1 TO n
630 PRINT
640 FOR i=1 TO n
650 PRINT z(k,i);" ";
660 NEXT
670 PRINT" ... ";b(k)
680 NEXT
690 PRINT
700 RETURN
710 REM iteration
720 ii=0
730 FOR k=1 TO n
740 sm(k)=b(k)
750 x(k)=0
760 NEXT
770 REM execution d'un  pas
780 GOSUB 70
790 PRINT "solution apres";ii;"boucles"
800 FOR k=1 TO n
810 x(k) =x(k)+a(0,k)
820 PRINT x(k)
830 NEXT
840 REM calcul des residus
850 PRINT"residus"
860 s1=0
870 FOR k=1 TO n
880 s=b(k)
890 FOR i=1 TO n
900 s=s-z(k,i)*x(i)
910 NEXT
920 sm(k)=s
930 PRINT s
940 s1=s1+s*s
950 NEXT
960 PRINT"moyenne quadratique:";s1/n
970 IF s1/n < 1E-12 THEN END
```

```
980 ii=ii+1
990 IF ii > 3 AND s1/n > 0.00001 THEN 1010
1000 GOTO 780
1010 PRINT"systeme incompatible ?":END
1020 REM matrice a par ligne
1030 DATA -74,80,18,-11,-4,-8
1040 DATA  14,-69,21,28,0,7
1050 DATA 66,-72,-5,7,1,4
1060 DATA -12,66,-30,-23,3,-3
1070 DATA 3,8,-7,-4,1,0
1080 DATA 4,-12,4,4,0,1
1090 REM second membre b
1100 DATA 0,0,0,1,0,0
```

```
systeme d'equations lineaires

-74   80   18  -11  -4  -8   ...  0

 14  -69   21   28   0   7   ...  0

 66  -72  -5   7   1   4   ...  0

-12   66  -30  -23   3  -3   ...  1

 3   8  -7  -4   1   0   ...  0

 4  -12   4   4   0   1   ...  0
orthogonalite solution= 6.21609E-08
orthogonalite solution= 9.70128E-11
solution apres 0 boucles
-39.9999428
 34.9999468
 154.999771
-1033.99848
-3210.99526
 4095.99397
residus
 1.52588E-05
-1.52588E-05
 0
 1.90735E-05
 0
-2.86102E-06
moyenne quadratique: 1.39607E-10
orthogonalite solution= 4.43951E-04
orthogonalite solution= 8.48862E-12
```

```
solution apres 1 boucles
-40.0004501
 35.0003988
 155.001754
-1034.0117
-3211.03634
 4096.04635
residus
 0
-7.62939E-06
 0
-7.62939E-06
 0
 0
moyenne quadratique: 1.94026E-11
orthogonalite solution= 7.37497E-04
orthogonalite solution= 4.54747E-12
solution apres 2 boucles
-40.0001449
 35.0001241
 155.000556
-1034.00371
-3211.01153
 4096.0147
residus
 0
 0
 0
 0
 0
 0
moyenne quadratique: 0


systeme d'equations lineaires

 3   2   ...  22

 2  -3   ... -22
orthogonalite solution= 2.79397E-09
orthogonalite solution= 5.82077E-11
solution apres 0 boucles
 1.69230769
 8.46153847
residus
-7.45058E-09
 0
moyenne quadratique: 2.77556E-17
```

```
systeme d'equations lineaires

 3   2   ...  22

 2  -3   ... -7
orthogonalite solution= 1.92085E-09
orthogonalite solution= 0
solution apres 0 boucles
 4
 5
residus
 7.45058E-09
-3.72529E-09
moyenne quadratique: 3.46945E-17
```

Changement de base

Ce petit programme permet de connaître l'expression d'un nombre donné en base 10, en base 2 ou en base 6. Il est basé sur une suite de divisions. Dans le cas du passage en base 16, remarquez l'astuce qui consiste à utiliser le code ASCII du caractère pour appeler A, B, C...

Voyons comment fonctionne le programme sur un exemple : conversion de 13 en base 2 :

$x = 13$ xx = INT (13/2) = 6
r = 3 − 2 = 1
$a\$$ = "1"
x = 6
$y\$$ = "1"

$x = 6$ xx = INT (6/2) = 3
r = 66 − 6 = 0
$a\$$ = "0"
x = 3
$y\$$ = "01"

$x = 3$ xx = INT (3/2) = 1
r = 3 − 2 = 1
$a\$$ = "1"
x =
$y\$$ = "101"

$x = 1$ xx = INT (1/2) = 0
r = 1
$a\$$ = "1"
x = 0
$y\$$ = "1101"

```
10 CLS
20 PRINT:PRINT"conversion d7un entier de base 10 e
n base..."
30 PRINT:INPUT"2 ou 16 ";b
40 PRINT:INPUT"nombre a convertir ";x
50 IF x=0 THEN END
55 IF b=16 THEN GOTO 150
60 REM
70 xx=INT(x/2)
80 r=x-2*xx
90 a$=STR$(INT(r))
100 x=xx
110 y$=a$+y$
120 IF  x>1 THEN GOTO 70
130 PRINT:PRINT Y$
140 REM
145 END
150 xx=INT(x/16)
160 r=INT(x-16*xx)
170 IF r<10 THEN a$=STR$(r) ELSE a$=CHR$(R+55)
180 x=xx
190 y$=a$+y$
200 IF x>1 THEN GOTO 150
210 PRINT:PRINT Y$
220 END
```

Logique

Pour les bricoleurs en microinformatique, il y a un problème qui se pose extrêmement souvent et qui est celui du décodage d'adresse relatif à un circuit (périphérique) relié à une unité centrale (microprocesseur). En effet, un microprocesseur 8 bits a un bus d'adresse constitué de 16 fils ; cela correspond à environ 65000 possibilités ; chaque circuit relié à ce bus doit être sélectionné par une adresse (au moins) présente sur ces fils. Il y a donc souvent une série de circuits logiques simples destinés à activer le périphérique (c'est-à-dire à lui appliquer « 1 » logique sur une entrée de sélection) lorsque l'adresse voulue est présente sur ces 16 fils. Le problème est donc : connaissant le circuit de décodage, quelles sont les adresses possibles du boîtier ?

En fait, un raisonnement souvent simple, permet de trouver la solution ; n'est-il pas plus amusant de faire un petit programme ? l'algorithme est simple ; il suffit d'examiner toutes les possibilités et de rechercher les bonnes solutions parmi elles. Enfin un programme qui fait faire à un engin idiot un travail idiot...

Premier exemple *(figure 1)*

Quelles sont les valeurs de A_0 . . . . . A_6, c'est-à-dire les adresses (sur 7 bits, donc 2 chiffres héxadécimaux) qui permettent d'obtenir L = 1 et M = 1 ?

On décompose le problème en 2 étapes :

– chercher les valeurs de H, I, J, K qui conviennent (lignes 40 à 140). Les différentes solutions sont rangées dans la matrice A. ZZ est leur nombre à la fin de cette étape.

– pour ces ZZ solutions, on examine maintenant les valeurs de A, B, C, D, E, F, G qui y conduisent. Ce seront les adresses convenables.

Remarque :

– Porte NAND	: L = NOT (A AND J AND I)
– Porte OUEXCLUSIF	: I = D XOR E
– Porte AND	: AND
– Porte OR	: OR
– Porte NOR	: NOT (– OR – OR –)
– "1" logique	: – 1
– "0" logique	: 0

– La solution s'exprime en héxadécimal par :

$$R = \sum_{i=0}^{i=n} 2^i . a_i \quad (\text{ici } i = 6)$$

```
20 DIM A(30,30)
40 FOR H=-1 TO 0
50 FOR J=-1 TO 0
60 FOR I=-1 TO 0
70 FOR K=-1 TO 0
80 L=NOT(H AND J AND I)
90 M=NOT(K)
100 IF NOT(L AND M) THEN 110
105 ZZ=ZZ+1:A(ZZ,1)=H
106 A(ZZ,2)=J:A(ZZ,3)=I:A(ZZ,4)=K
110 NEXT K
120 NEXT I
130 NEXT J
140 NEXT H
150 REM SECONDE ETAPE
160 FOR yy=1 TO zz
170 FOR A=-1 TO 0
180 FOR B=-1 TO 0
190 FOR C=-1 TO 0
200 FOR D=-1 TO 0
210 FOR E=-1 TO 0
220 FOR F=-1 TO 0
230 FOR G=-1 TO 0
240 h=NOT( a AND b AND c) :i=d XOR e
250 j=NOT(f OR g):k=j
260 IF (h=a(yy,1)) AND (i=a(yy,2)) AND ( j=a(yy,3)
) AND ( k=a(yy,4)) THEN r=64*ABS(g)+32*ABS(f)+16*A
BS(e)+8*ABS(d)+4*ABS(c)+2*ABS(b)+ABS(a):PRINT HEX$
(r,2);" ";
270 NEXT g:NEXT f:NEXT e:NEXT d:NEXT c:NEXT b:NEXT
 a
280 NEXT yy
```

```
6B 2B 4B 73 33 53 6D 2D 4D 75 35 55 69 29 49 71 31
 51 6E 2E 4E 76 36 56 6A 2A 4A 72 32 52 6C 2C 4C
74 34 54 68 28 48 70 30 50 7B 3B 5B 63 23 43 7D 3D
 5D 65 25 45 79 39 59 61 21 41 7E 3E 5E 66 26 46
7A 3A 5A 62 22 42 7C 3C 5C 64 24 44 78 38 58 60 20
 40 6F 2F 4F 77 37 57 7F 3F 5F 67 27 47
```

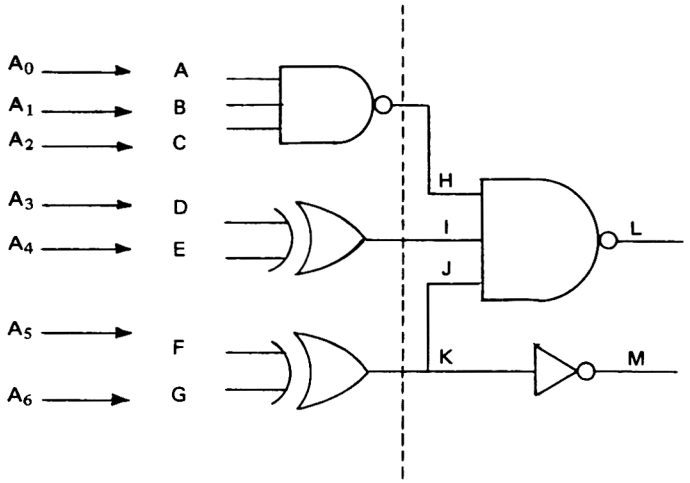


Fig. 1.

Second exemple

Celui-ci est extrait de la notice d'un kit d'évaluation pour microprocesseur de la firme HEATHKIT (marque déposée). Les lignes d'adresse A_{15} à A_3 sont reliées au décodeur d'adresse constitué par 3 portes NOR (IC – 2A, IC – 2B, IC – 2C) et une porte NAND à 8 entrées (IC – 3). Les lignes A_0, A_1, A_2 permettent de sélectionner 7 registres internes de IC – 1. Celui-ci est validé par E = 0, donc par une sortie de IC – 1 à "0" (appelant $\overline{K}$ cette sortie, il faut donc avoir K = 1).

Le programme évalue d'abord les sorties de IC – 2, appelées I, H, G ; K est aussitôt évalué et s'il vaut "1", la solution est affichée en héxadécimal. Une adresse seule convient ici : c'est 02F0 ; les 8 registres internes sont donc validés par les adresses 02F0 à 02F7.

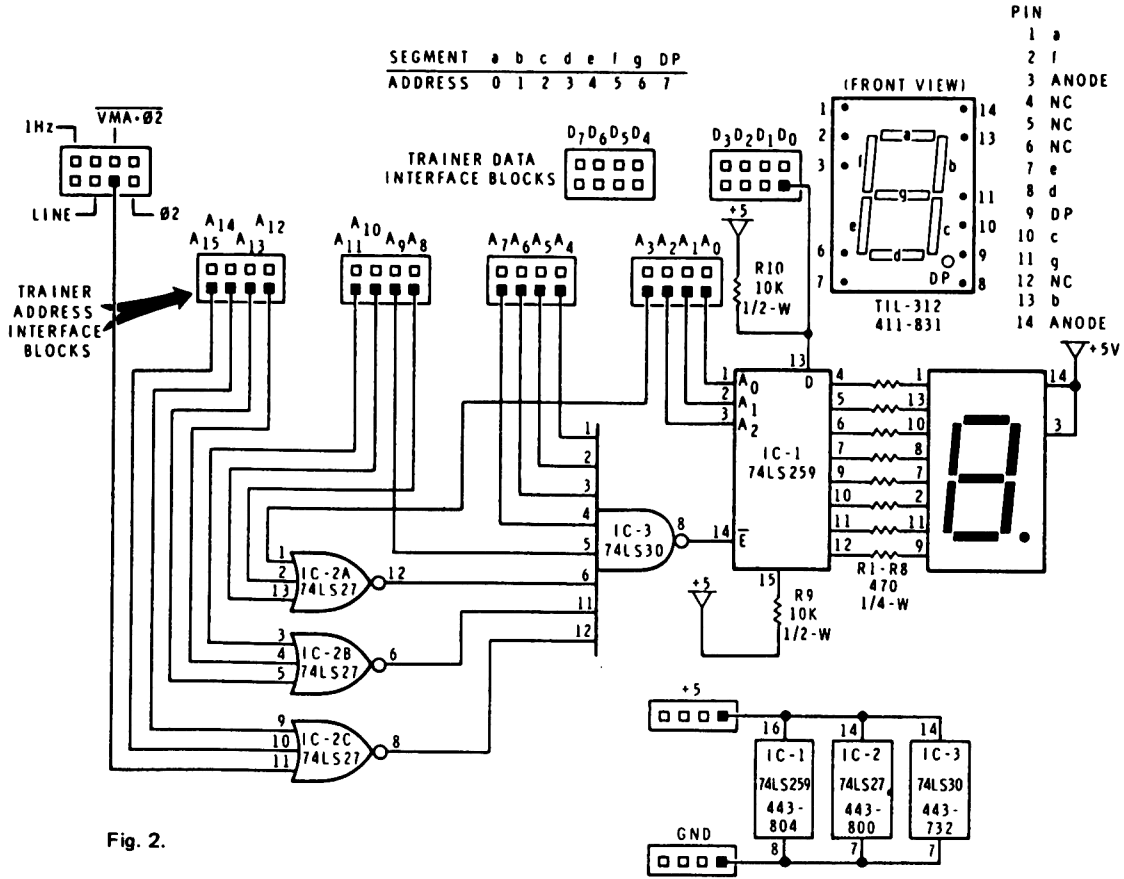

SEGMENT a b c d e f g DP
ADDRESS 0 1 2 3 4 5 6 7
1Hz
VMA·Ø2
LINE
Ø2
TRAINER ADDRESS INTERFACE BLOCKS
TRAINER DATA INTERFACE BLOCKS
D7D6D5D4
D3D2D1D0
A15 A14 A13 A12
A11 A10 A9A8
A7A6A5A4
A3A2A1A0
+5
R10 10K 1/2-W
(FRONT VIEW)
TIL-312 411-831
PIN
1 a
2 f
3 ANODE
4 NC
5 NC
6 NC
7 e
8 d
9 DP
10 c
11 g
12 NC
13 b
14 ANODE
+5V
IC-1 74LS259
IC-3 74LS30
IC-2A 74LS27
IC-2B 74LS27
IC-2C 74LS27
R1-R8 470 1/4-W
R9 10K 1/2-W
GND
IC-1 74LS259 443-804
IC-2 74LS27 443-800
IC-3 74LS30 443-732

Fig. 2.

```
10 FOR A3=
-1 TO 0
20 FOR A4=-1 TO 0
30 FOR A5=-1 TO 0
40 FOR A6=-1 TO 0
50 FOR A7=-1 TO 0
60 FOR A8=-1 TO 0
70 FOR A9=-1 TO 0
80 FOR A=-1 TO 0
90 FOR B=-1 TO 0
110 FOR C=-1 TO 0
120 FOR D=-1 TO 0
130 FOR E=-1 TO 0
140 FOR F=-1 TO 0
150 FOR VMA =-1 TO 0
160 I=NOT(E OR F OR VMA )
170 H=NOT(B OR C OR D)
180 G=NOT(A3 OR A8 OR A)
190 K=A4 AND A5 AND A6 AND A7 AND A9 AND G AND H A
ND I
200 IF K THEN R=ABS(A0)+2*ABS(A1)+4*ABS(A2)+8*ABS(
A3)+16*ABS(A4)+32*ABS(A5)+64*ABS(A6)+128*ABS(A7)+2
56*ABS(A8)+512*ABS(A9)+1024*ABS(A)+2048*ABS(B)+409
6*ABS(C)+8192*ABS(D)+16384*ABS(E)+32768*ABS(F):PRI
NT HEX$(R,4)
210 NEXT VMA
220 NEXT F
230 NEXT E
240 NEXT D
250 NEXT C
260 NEXT B
270 NEXT A
280 NEXT A9
290 NEXT A8
300 NEXT A7
310 NEXT A6
320 NEXT A5
330 NEXT A4
340 NEXT A3
```

```
RUN

02F0
```

Troisième exemple *(figure 3)*

Quelles sont les adresses possibles de IC — 1 ? (*Attention :* A_{10} à A_4 ne sont pas employées).

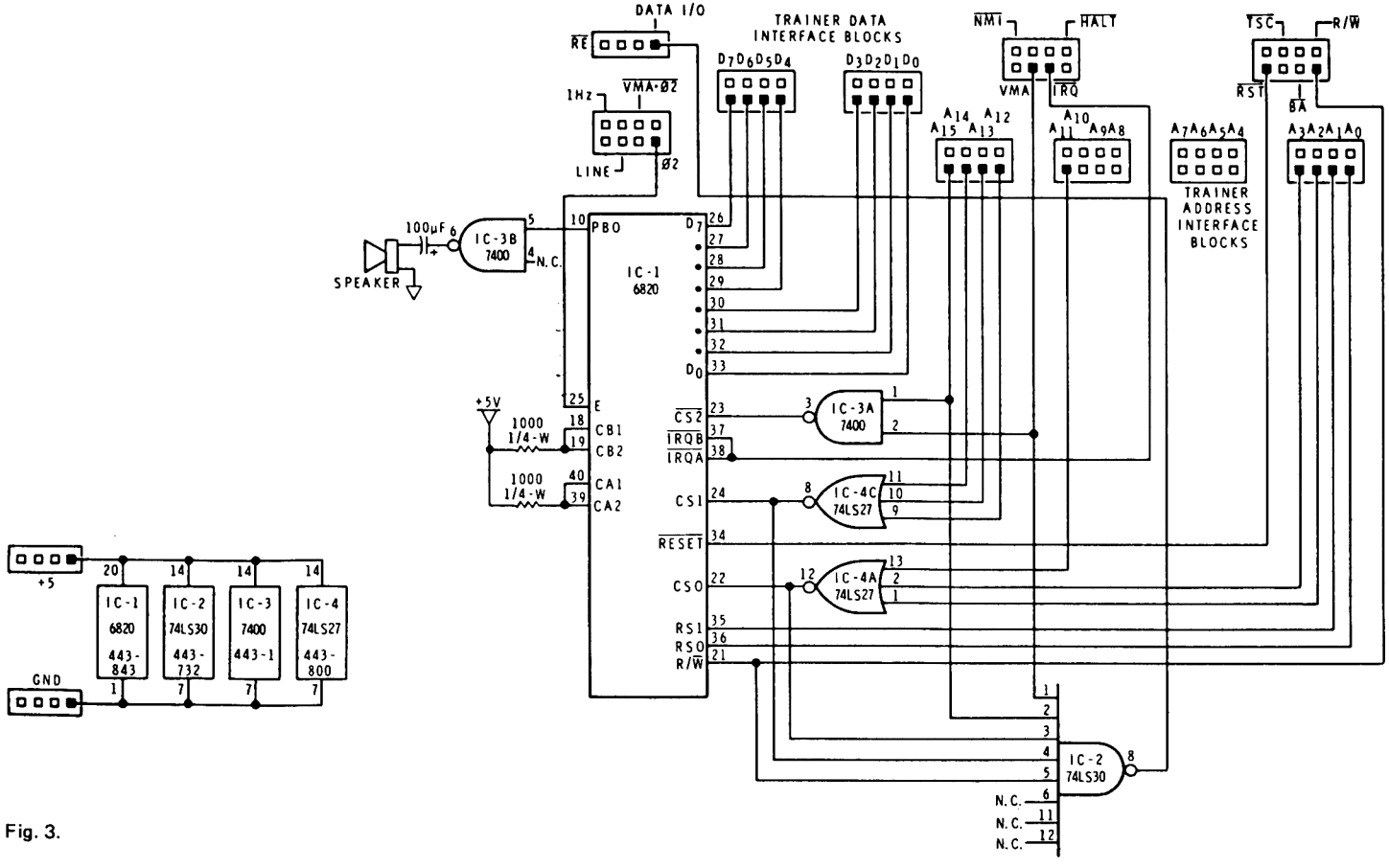

DATA I/O
RE
TRAINER DATA INTERFACE BLOCKS
D7D6D5D4
D3D2D1D0
NMI
HALT
VMA
IRQ
TSC
R/W
RST
BA
1Hz
VMA·Ø2
LINE
Ø2
A14 A13
A15
A12
A11 A9A8
A10
A7A6A5A4
A3A2A1A0
TRAINER ADDRESS INTERFACE BLOCKS
100µF
IC-3B 7400
N.C.
SPEAKER
IC-1 6820
PBO
D7
D0
E
CB1
CB2
CA1
CA2
CS2
IRQB
IRQA
CS1
RESET
CS0
RS1
RS0
R/W
+5V
1000 1/4-W
1000 1/4-W
IC-3A 7400
IC-4C 74LS27
IC-4A 74LS27
IC-2 74LS30
N.C.
+5
GND
IC-1 6820 443-843
IC-2 74LS30 443-732
IC-3 7400 443-1
IC-4 74LS27 443-800

Fig. 3.

Séries de Fourier

Un signal périodique, de période T, est décomposable en série de Fourier :

$$f(t) = a_0 + \sum_{n=1}^{n=\infty} a_n \cos n\omega t + b_n \sin n\omega t$$

$$a_0 = \frac{1}{T}\int_0^T f(t)\,dt$$

$$a_n = \frac{2}{T}\int_0^T f(t) \cos n\omega t\,dt$$

$$b_n = \frac{2}{T}\int_0^T f(t) \sin n\omega t\,dt$$

Si l'on connaît la période T, il est possible de calculer numériquement ces intégrales.

Si $f(t)$ est connue sur N points $f(0)$, $f\left(\frac{T}{N}\right)$, $f\left(\frac{2T}{N}\right)$, ... $f\left(\frac{(N-1)T}{N}\right)$, l'intégrale donnant a_n devient :

$$a_n = \frac{2}{T} \sum_{i=0}^{i=N-1} f\left(\frac{iT}{N}\right) \cos\left(n\omega \frac{iT}{N}\right) \cdot \frac{T}{N}$$

Pour accélérer les calculs, on peut poser :

$$A = \frac{n\omega T}{N} = \frac{n2\pi T}{NT} = \frac{2\pi n}{N}$$

Nous allons utiliser cette méthode pour calculer les séries de Fourier de quelques signaux. Nous découvrirons au passage les limites d'utilisation de cette méthode et la manière d'interpréter les résultats qu'elle fournit.

Tous les programmes proposés sont calqués sur le même modèle, aussi ne commenterons-nous que le premier complètement.

Les deux premiers concernent le calcul d'une série de Fourier : calcul numérique et représentation graphique d'un spectre.

Le troisième montre l'influence sur le spectre d'un signal de la transmittance d'un circuit.

Les quatrième et cinquième programmes démontrent l'utilisation qui peut être faite d'une série de Fourier : évaluer graphiquement la tension de sortie d'un filtre passe-bas du premier et du second ordre.

Programme 1

n points de calcul
nh nombre d'harmoniques calculés
j variable de boucle (varie de 0 à nh)
a_n coefficient du terme en cosinus correspondant à l'harmonique j
b_n coefficient du terme en sinus correspondant à l'harmonique j

Il permet d'afficher les valeurs numériques des coefficients a_n et b_n. Nous donnons à titre de comparaison les valeurs exactes de ces coefficients pour un signal carré. On peut également étudier le module de chaque terme, soit SQR $(a_n^2 + b_n^2)$.

Pour un signal carré, on a :

$$f(t) = 1 \quad \text{pour } 0 < t < \frac{T}{2}$$

$$f(t) = -1 \quad \text{pour } \frac{T}{2} < t < T$$

$$a_n = 0 \quad \text{quelque soit } n$$

$$b_n = \frac{4}{\pi n} \quad \text{pour } n \text{ impair, sinon } 0.$$

On constate que la précision augmente quand on prend un plus grand nombre de termes (n grand).

```
10 REM ********************
20 REM *                  *
30 REM * serie de fourier *
40 REM *                  *
50 REM *   programme 1    *
60 REM *                  *
70 REM ********************
80 REM
90 CLS
100 PRINT:PRINT"entrez la fonction en ligne 110"
110  DEF FNf(i)=SGN(SIN(2*PI*i/n))
120 INPUT " nombre de points de calcul:";n
130 INPUT "harmoniques calcules :";nh
140 FOR j=0 TO nh
150 an=0:bn=0
160 a=2*PI*J/n
170 FOR i=0 TO n-1
180 an=FNf(i)*COS (a*i) +an
190 bn=FNf(i)*SIN (a*i) +bn
200 NEXT i
210 bn=2*bn/n
220 an=2*an/n
230 GOTO 250
240 PRINT "a";J;"=";an,"b";J;"=";bn
250 PRINT "nh..";j;"...->  ";SQR(an^2+bn^2)
260 NEXT j
```

```
 FNf(i)=SGN(SIN(2*PI*i/n))
 nombre de points de calcul: 32
harmoniques calcules : 10
nh.. 0 ...->   0
nh.. 1 ...->   1.2691463
nh.. 2 ...->   8.59056E-10
nh.. 3 ...->   0.412069776
nh.. 4 ...->   4.82193E-10
nh.. 5 ...->   0.233858552
nh.. 6 ...->   1.19656E-09
nh.. 7 ...->   0.152312941
nh.. 8 ...->   4.11590E-11
nh.. 9 ...->   0.102584849
nh.. 10 ...->   2.18279E-11
```

```
 FNf(i)=SGN(SIN(2*PI*i/n))
 nombre de points de calcul: 128
harmoniques calcules : 10
nh.. 0 ...->   0
nh.. 1 ...->   1.27298387
nh.. 2 ...->   3.14317E-10
nh.. 3 ...->   0.423645914
nh.. 4 ...->   1.40878E-10
nh.. 5 ...->   0.253368307
nh.. 6 ...->   1.10900E-09
nh.. 7 ...->   0.180098188
nh.. 8 ...->   4.53089E-10
nh.. 9 ...->   0.139162571
nh.. 10 ...->   3.33525E-10


valeurs exactes
..nh  1 ...->  1.27323954
..nh  3 ...->  0.424413182
..nh  5 ...->  0.254647909
..nh  7 ...->  0.181891364
..nh  9 ...->  0.141471061
..nh  11 ...->  0.11574905
```

Programme 2

Il est identique au précédent jusqu'à la ligne 260. A partir de là, l'amplitude des harmoniques est stockée dans une matrice Y (*nh*). La valeur maximum *mx* des coefficient est évaluée. Puis le spectre du signal est tracé tout en étant normalisé (la plus grande raie aura toujours 400 points de hauteur).

Nous proposons un exemple particulier et pour lequel théoriquement le théorème de Fourier ne s'applique pas. En effet, sur la période T, nous avons construit (ligne 110) une fonction sinusoïdale de période $\frac{T}{2,5}$. Comme le calcul ne peut mettre en évidence que les coefficients correspondant aux fréquences $\frac{1}{T}, \frac{2}{T}, \frac{3}{T}, \ldots$, il ne peut évidemment nous donner une raie à $\frac{2,5}{T}$. Pourtant, on constate que le spectre (faux) obtenu est à peu près centré sur $\frac{2,5}{T}$ et que finalement ces résultats sont exploitables. Nous reviendrons plus loin sur ce phénomène.

D'autre part, et seconde constatation indépendante de la précédente, que se passe-t-il si l'on cherche à évaluer les harmoniques « lointaines », pour lesquels le calcul ne devrait pas être fait ; en effet, il y a alors trop peu de points et l'équivalence : calcul de l'intégrale = calcul numérique n'est plus justifiée. La seconde figure montre le résultat : le spectre obtenu est symétrique par rapport à $\frac{n}{2T}$. Résultat remarquable ! En effet, sur une période T, nous prenons n valeurs de la fonction étudiée. Cela suppose implicitement que cette fonction est nulle le reste du temps.

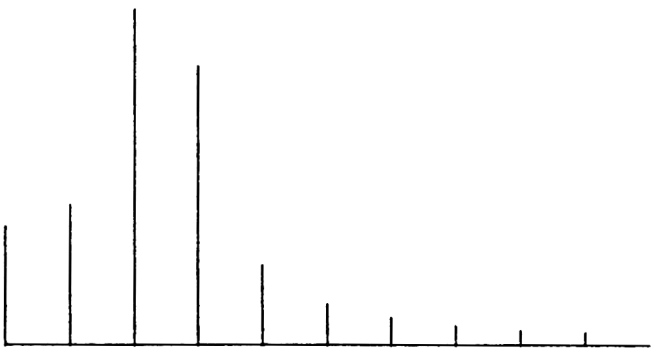
Conclusion : nous pensons effectuer le calcul approché des termes du développement, mais nous effectuons le calculexact de ces termes pour une fonction définie seulement en quelques points. Et le spectre obtenu est justement celui-là, qui ressemble dans sa moitié gauche à celui qui est cherché.

Il y a ce que l'on appelle « recouvrement » ; c'est-à-dire que les raies de fréquence supérieure à $\frac{n}{2T}$ viennent se mélanger aux raies de fréquence inférieure. En conclusion, le résultat (le spectre de 0 à $\frac{n}{2T}$) ne peut être exact que si le signal de départ n'a pas d'harmoniques de fréquence supérieure à $\frac{n}{2T}$: c'est le théorème de Shannon.

```
10 REM ********************
20 REM *                  *
30 REM * serie de fourier *
40 REM *                  *
50 REM *   programme 2    *
60 REM *                  *
70 REM ********************
80 REM
90 CLS
100 PRINT"entrez la fonction en ligne 110"
110  DEF FNf(i)=SIN(2*PI*i*2.5/n)
120 mx=0
130 INPUT " nombre de points de calcul:";n
140 INPUT "harmoniques calcules :";nh
150 DIM y(nh)
160 FOR j=0 TO nh
170 an=0:bn=0
180 a=2*PI*J/n
190 PRINT "j=";j
```

```
200 FOR i=0 TO n-1
210 an=FNf(i)*COS (a*i) +an
220 bn=FNf(i)*SIN (a*i) +bn
230 NEXT i
240 bn=2*bn/n
250 an=2*an/n
260 y(j)=SQR(an*an+bn*bn)
270 IF J=0 THEN 290
280 IF y(j)>y(j-1) THEN mx=y(j)
290 NEXT j
300 MODE 2
310 PLOT 0,2:DRAW 640,2
320 FOR J=0 TO nh
330 PLOT J*639/nh,0:DRAWR 0,y(J)*399/mx
340 NEXT J
```

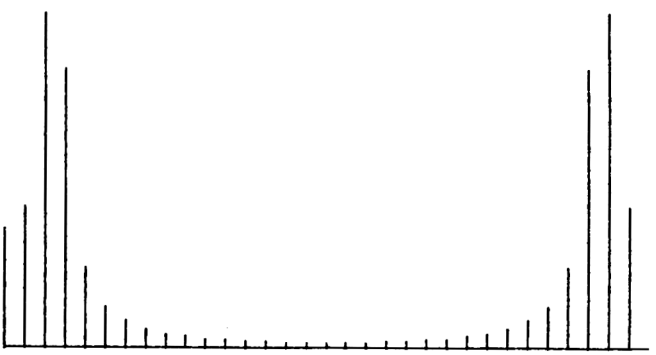
FNf(i)=SIN(2*PI*i*2.5/n)
nombre de points de calcul: 32
harmoniques calcules : 10



FNf(i)=SIN(2*PI*i*2.5/n)

nombre de points de calcul: 32

harmoniques calcules : 32



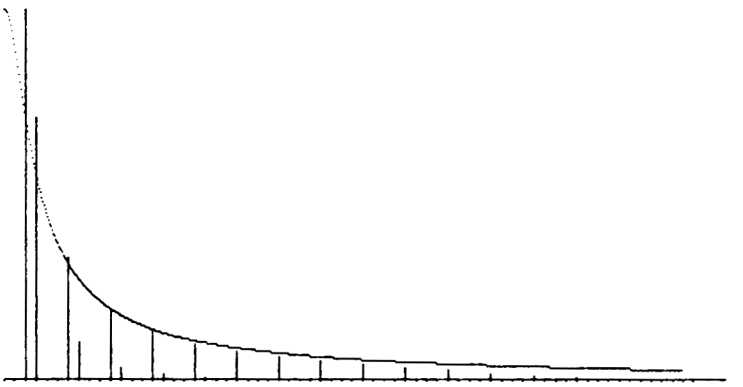
Programme 3

Il montre l'exploitation qui peut être faite du théorème de Fourier. Un signal (carré en l'occurence), dont le spectre est calculé (et n'est retenu que jusqu'à l'ordre $\frac{n}{2}$ ligne 150 pour les raisons qui précèdent), traverse un filtre RC du premier ordre, dont la transmittance (lignes 370 - 390) est tracée point par point. Le programme déduit le spectre du signal de sortie, (légèrement décalé pour des raisons de visibilité).

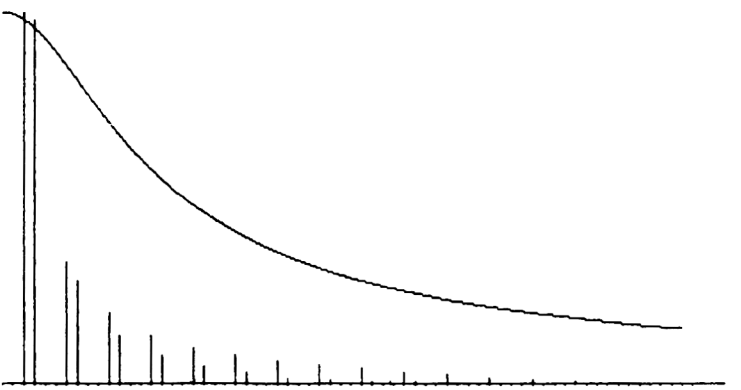
```
10 REM ********************
20 REM *                  *
30 REM * serie de fourier *
40 REM *                  *
50 REM *   programme 3    *
60 REM *                  *
70 REM ********************
80 REM
90 CLS
100 PRINT"entrez la fonction en ligne 110"
110  DEF FNf(i)=SGN(SIN(2*PI*i/n))
120 mx=0
130 INPUT " nombre de points de calcul:";n
140 INPUT "rapport f/fc :";k
150 nh=n/2
160 DIM y(nh)
170 FOR j=0 TO nh
180 an=0:bn=0
190 a=2*PI*J/n
200 PRINT "j=";j
210 FOR i=0 TO n-1
220 an=FNf(i)*COS (a*i) +an
230 bn=FNf(i)*SIN (a*i) +bn
240 NEXT i
250 bn=2*bn/n
260 an=2*an/n
270 y(j)=SQR(an*an+bn*bn)
280 IF y(j)>mx THEN mx=y(j)
290 NEXT j
300 MODE 2
310 PLOT 0,2:DRAW 640,2
320 FOR J=0 TO nh
330 s=y(j)/SQR(1+(j*k)^2)
```

```
340  PLOT (J+0.5)*600/nh,0:DRAWR 0,s*399/mx
350  PLOT J*600/nh,0:DRAWR 0,y(J)*399/mx
360 NEXT J
370 FOR J=0 TO nh  STEP nh/600
380 PLOT J*600/nh,399/SQR(1+(j*k)^2)
390 NEXT
```

FNf(i)=SGN(SIN(2*PI*i/n))
nombre de points de calcul: 64
rapport f/fc : 1



FNf(i)=SGN(SIN(2*PI*i/n))
nombre de points de calcul: 64
rapport f/fc : 0.2

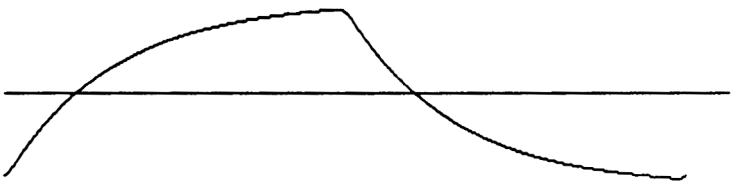


Programme 4

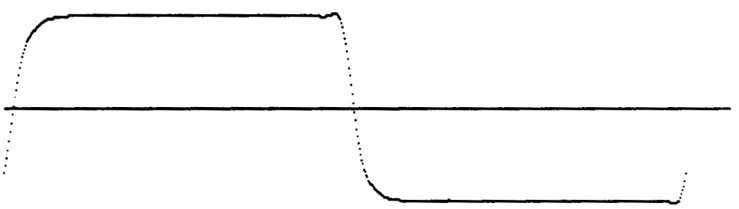
Il permet, dans les mêmes conditions, de tracer la courbe donnant l'aspect du signal de sortie de ce filtre, en tenant compte de l'atténuation et du déphasage introduits par celui-ci sur chaque harmonique.

```
10 REM ********************
20 REM *                  *
30 REM * serie de fourier *
40 REM *                  *
50 REM *   PROGRAMME 4    *
60 REM *                  *
70 REM ********************
80 REM
90 CLS
100 PRINT"entrez la fonction en ligne 110"
110  DEF FNf(i)=SGN(SIN(2*PI*i/n))
120 REM -> la fonction doit etre impaire
130 mx=0
140 INPUT " nombre de points de calcul:";n
150 INPUT "rapport f/fc :";k
160 nh=n/2
170 DIM y(nh)
180 FOR j=0 TO nh
190 an=0:bn=0
200 a=2*PI*J/n
210 PRINT "j=";j
220 FOR i=0 TO n-1
230 bn=FNf(i)*SIN (a*i) +bn
240 NEXT i
250 bn=2*bn/n
260 y(j)=bn
270 NEXT j
280 MODE 2
290 PLOT 0,200:DRAW 640,200
300 FOR t=0 TO 600
310 FOR J=0 TO nh
320 s=y(j)*SIN(2*PI*j*t/600-ATN(j*k))/SQR(1+(j*k)^2)
330 ss=ss+s
340 NEXT j
350 PLOT t,200+100*ss
360 ss=0
370 NEXT t
```

```
nombre de points de calcul: 64
rapport f/fc : 1
```



```
nombre de points de calcul: 64
rapport f/fc : 0.1
```



Programme 5

Il réalise la même opération pour un filtre du second ordre, pour lequel le rapport F/F_0 (fondamental sur fréquence naturelle) et le coefficient d'amortissement *m* sont donnés.

Si, avec la méthode utilisée dans les programmes précédents, et avec les précautions d'emploi énumérées, il est possible d'évaluer une série de Fourier, il n'en reste pas moins que le temps de calcul devient vite prohibitif dès que *n* est assez grand, disons supérieur à 64. En effet, doubler *n* permet d'évaluer à 2 fois plus de raies, au prix d'un plus grand

nombre de multiplications pour l'évaluation de chaque intégrale. En fait, à n donné, il faut n multiplications par raie du spectre, d'où environ un temps de calcul proportionnel à n^2.

Deux mathématiciens, Cooley et Tukey (1965) ont réussi à optimiser ces calculs et ont mis au point un algorithme, applicable lorsque n est une puissance de 2, dont le temps de calcul est proportionnel à $n \mathrm{Log} n$, en se servant de symétries qui interviennent pour ce choix de n. Cette méthode sera développée dans le chapitre suivant.

```
10 REM ********************
20 REM *                  *
30 REM * serie de fourier *
40 REM *                  *
50 REM *   PROGRAMME 5    *
60 REM *                  *
70 REM ********************
80 REM
90 CLS
100 PRINT"entrez la fonction en ligne 110"
110  DEF FNf(i)=SGN(SIN(2*PI*i/n))
120 REM -> la fonction doit etre impaire
130 mx=0
140 INPUT " nombre de points de calcul:";n
150 INPUT "rapport f/fc :";k
160 INPUT "coeff. d amortissement:";m
170 nh=n/2
180 DIM y(nh)
190 DIM t(nh):DIM p(nh)
200 FOR j=0 TO nh
210 r=1-(j*k)^2: i=2*m*j*k
220 t(j)=1/SQR(r*r+i*i)
230 IF r=0 THEN p(j)=PI/2:GOTO 250
240 p(j)=ATN(i/r)
250 IF p(j) < 0 THEN p(j)=PI+p(j)
260 bn=0
270 a=2*PI*J/n
280 PRINT "j=";j
290 FOR i=0 TO n-1
300 bn=FNf(i)*SIN (a*i) +bn
310 NEXT i
320 bn=2*bn/n
330 y(j)=bn
340 NEXT j
350 MODE 2
360 LOCATE 5,20:PRINT"f/fc=";k
370 LOCATE 5,21 : PRINT"m=";m
```

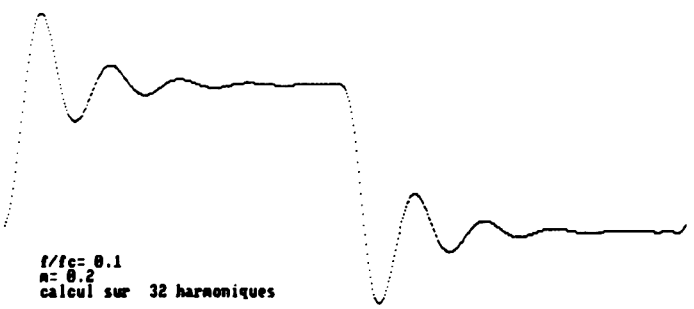
```
380 LOCATE 5,22 : PRINT"calcul sur ";nh ;"harmonique
390 FOR t=0 TO 600
400 FOR J=0 TO nh
410 s=y(j)*t(j)*SIN(2*PI*j*t/600-p(j))
420 ss=ss+s
430 NEXT j
440 PLOT t,200+80*ss
450 ss=0
460 NEXT t
```

```
FNf(i)=SGN(SIN(2*PI*i/n))
 nombre de points de calcul: 64
  0  1.70141E+38
```

```
FNf(i)=SGN(SIN(2*PI*i/n))
 nombre de points de calcul: 64
rapport f/fc : 0.1
 0  0  0
```

```
FNf(i)=SGN(SIN(2*PI*i/n))
 nombre de points de calcul: 64
rapport f/fc : 0.1
coeff. d amortissement: 0.2
```

```
FNf(i)=SGN(SIN(2*PI*i/n))
 nombre de points de calcul: 64
rapport f/fc : 0.1
coeff. d amortissement: 0.2
```

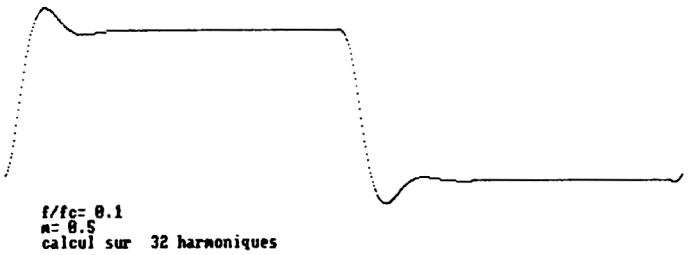


FNf(i)=SGN(SIN(2*PI*i/n))

nombre de points de calcul: 64

rapport f/fc : 0.1

coeff. d amortissement: 0.5

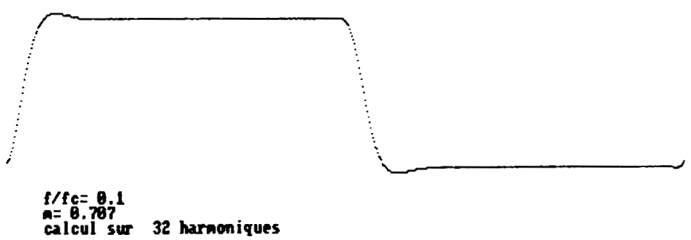


FNf(i)=SGN(SIN(2*PI*i/n))

nombre de points de calcul: 64

rapport f/fc : 0.1

coeff. d amortissement: 0.707



Transformée de Fourier rapide

La transformation de Fourier est un outil très puissant dans l'étude des signaux de toute sorte.

Le théorème de Fourier proprement dit peut faire correspondre à une fonction continue et périodique $f(t)$ (de période T) le développement :

$$f(t) = A_0 + \sum_{N=1}^{N=\infty} A_n \cos n\omega t + \sum_{N=1}^{N=\infty} Bn \sin n\omega t$$

où l'on a :

$$A_0 = \frac{1}{T}\int_0^T f(t)\,dt$$

$$B_n = \frac{2}{T}\int_0^T f(t) \sin n\omega t\,dt \qquad A_n = \frac{2}{T}\int_0^T f(t) \cos n\omega t\,dt$$

Des expressions équivalentes sont les suivantes :

- $f(t) = A_0 + \sum_{n=1}^{n=\infty} \alpha_n \cos(n\omega t + \varphi_n)$

- $f(t) = \sum_{n=-\infty}^{n=+\infty} C_n \exp(jn\omega t)$

Cette dernière expression fait apparaître des fréquences « négatives », dont la présence se justifie par le fait que $f(t)$ est réelle. Cela a pour conséquence l'existence de symétries, utiles pour la compréhension des phénomènes étudiés plus loin.

- On a $\alpha_n^2 = a_n^2 + b_n^2$ $\qquad \varphi_n = -\text{Arctg}\left(\frac{b_n}{a_n}\right)$

- On a $C_n = \int_{-T/2}^{T/2} f(t) . \exp(-jn\omega t)\, dt = a_n - jb_n = C^*_{-n}$

Ceci permet de faire correspondre au signal $f(t)$ — périodique rappelons-le — une suite de couples de nombres :

0	α_0
1	α_1
2	α_2

qui sont les amplitudes des raies du spectre de $f(t)$; ce spectre contient ainsi les pulsations 0 (le «continu»), 1 (le fondamental), 2, 3, ...

Que se passe-t-il si $f(t)$ n'est pas périodique (soit si $T \to \infty$) ? On peut voir que les relations de définition des coefficients deviennent des intégrales et par conséquent des fonctions *continues* de ω.

On associe alors à $f(t)$ la fonction $F(\omega)$ définie par exemple par :

$$F(\omega) = \int_{-\infty}^{+\infty} f(t) . \exp(-j\omega t)\, dt$$

$F(\omega)$ est la transformée de Fourier de $f(t)$.

L'échantillonnage d'un signal F (t)

Considérons une fonction continue du temps $f(t)$ et supposons que nous en notions les valeurs à des intervalles de temps égaux : Te.

Au signal continu $f(t)$, correspond maintenant un ensemble de valeurs numériques ($f(0)$, $f(Te)$, $f(2Te)$ ...).

On démontre que ces deux notions sont identiques si la plus grande fréquence contenue dans le spectre de $f(t)$ est inférieure à $\frac{1}{2Te} = \frac{Fe}{2}$ (Théorème de Shannon).

Supposons maintenant $f(t)$ quelconque et tentons à l'aide d'un calculateur numérique d'évaluer la transformée de Fourier de ce signal $f(t)$. C'est possible puisque nous possédons des nombres (les valeurs numériques de $f(t)$ aux instants d'échantillonnage) et nous pouvons donc évaluer numériquement ces intégrales.

Mais un problème surgit : il n'est pas possible d'évaluer ces intégrales de moins l'infini à plus l'infini. Aussi est-t-on amené à limiter le nombre d'échantillons à un nombre fini N.

Cette limitation va introduire des erreurs : tout se passe comme si l'on multipliait la fonction $f(t)$ par une « fenêtre », c'est-à-dire une fonction valant 1 de 0 à $NT_e = T$ et 0 partout ailleurs. Le spectre obtenu par une transformée de Fourier discrète (celle que nous tentons de réaliser) sera donc celui fourni par une « vraie » transformée, qui correspond au produit de $f(t)$ par la fenêtre. On peut réaliser celle-ci pour savoir quel en sera le résultat. Celui-ci est simple : on reporte en chaque point de la « vraie » transformée de $f(t)$ la transformée de la fenêtre et l'on effectue la somme des courbes obtenues (cette démarche s'appelle : convoluer les 2 fonctions).

D'autre part, et ceci est toujours vrai, échantillonner un signal produit un spectre aussi par convolution : le spectre du signal original est reporté sur fe, $2\,fe$, . . ., (on voit ici l'intérêt des fréquences « négatives ») le spectre résultant étant la somme de tous ces spectres. Ainsi le spectre original n'est pas déformé (entre 0 et $Fe/2$) si effectivement la fonction étudiée ne contient pas de fréquences supérieures à $Fe/2$.

Enfin, cherchant numériquement une fonction $F(\omega)$ continue, il ne nous sera possible de n'obtenir que quelques valeurs de celles-ci. Il est classique à l'heure actuelle d'effectuer une Fast Fourier Transform pour laquelle :

— le nombre d'échantillons N est une puissance de 2 (4, 8, 16, 32, 64, ...).

— les fréquences pour lesquelles $F(\omega)$ est calculée sont $0, \frac{1}{T}, \frac{2}{T}, \frac{3}{T}, \ldots, \frac{N-1}{T}$, en appelant T la durée de la fenêtre.

— la fréquence d'échantillonnage étant par conséquent $\frac{N}{T}$, le spectre obtenu sera toujours symétrique par rapport à $\frac{N}{2T}$. Seule la moitié des raies du spectre apporteront une information (dans la mesure où il n'y aura pas de recouvrement d'ailleurs, c'est-à-dire mélange des raies du spectre de $f(t)$ avec les raies parasites créées par l'échantillonnage).

Le problème de la fenêtre

Tout se passerait fort bien si dans la fenêtre de durée T, il y avait un nombre entier de périodes des sinusoïdes correspondant à $\frac{1}{T}, \frac{2}{T}, \frac{3}{T}, \ldots$.

Malheureusement, ce ne peut toujours être le cas (signal non périodique, par exemple).

Que se passe-t-il alors dans le cas contraire ?

Un retour à la méthode de convolution nous l'apprend : des raies parasites vont naître et fausser le diagnostic. Les mathématiciens proposent comme remèdes d'autres types de fenêtres dont les ondulations sont beaucoup plus faibles que pour la fenêtre rectangulaire, mais dont malheureusement la largeur du lobe est plus grande.

Dans la fenêtre	Dans le spectre
– 1 période de sinusoïde	– 1 raie à $\frac{1}{T}$
– 2 périodes de sinusoïde	– 1 raie à $\frac{2}{T}$
– 3 périodes de sinusoïde	– 1 raie à $\frac{3}{T}$
.	.
.	.
.	.
.	.

On pourra constater sur les dessins l'influence de la fenêtre sur le spectre obtenu.

Dans la figure 1, la fenêtre de largeur T contient un signal périodique, de fréquence fondamentale 1/T. Une seule raie apparaît alors dans le spectre, et c'est la bonne.

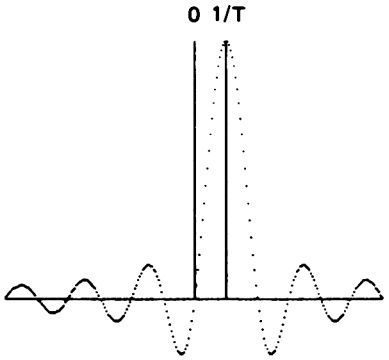
Dans la figure 2, nous avons supposé que la fenêtre contenait 1,5 périodes d'un signal. Ceci se traduirait normalement par une raie dans le spectre à 1.5/T. Mais la FFT ne peut mettre en évidence une telle raie ; elle ne calcule en effet que l'amplitude des raies présentes en 1/T, 2/T, ...

L'utilisation de la fenêtre rectangulaire se traduit alors par l'apparition de fausses raies en 1/T, 2/T,.

En utilisant un autre type de fenêtre, dont le spectre est plus « lourd » mais d'ondulations beaucoup plus faibles, il est possible, au détriment de la qualité du spectre d'un signal pur (figure 3), de réorganiser celui d'un signal de fréquence 1.5/T (figure 4).

Cela est du au fait que la FFT ne sait pas que le signal a sa propre période ; elle considère que celle-ci est T dans tous les cas.

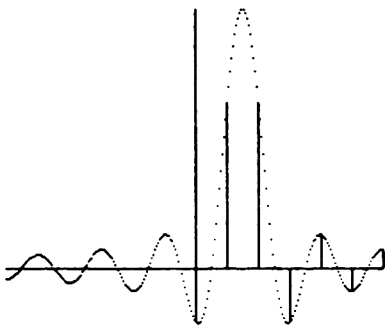
Ainsi, le spectre observé n'est pas celui du signal proprement dit, mais d'un signal qui a la même forme que celui étudié, mais qui serait périodique de période T.



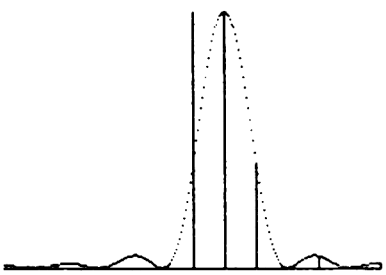
① En continu : la raie à $\frac{1}{T}$ (sinusoïde pure).

En pointillé : le spectre de la fenêtre rectangulaire.

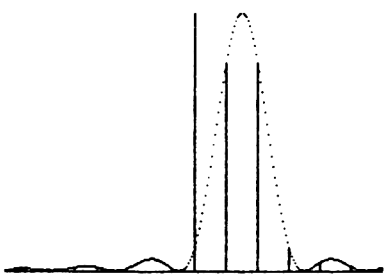
La convolution de ces 2 fonctions ne génère aucune raie à $\frac{2}{T}$, $\frac{3}{T}$, ...



② Une raie à $\frac{1,5}{T}$, non mise en évidence par la FFT génère des raies parasites lors de la convolution avec le spectre de la fenêtre.



③ Idem à ①, mais fenêtre de Blackmann.

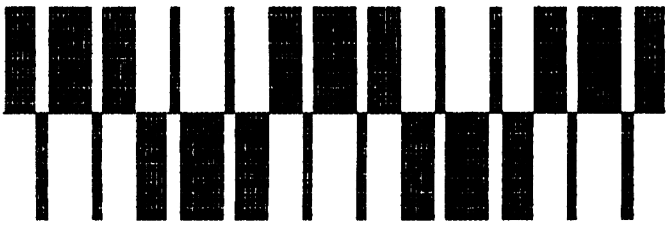


④ Idem à ②, mais fenêtre de Blackmann.

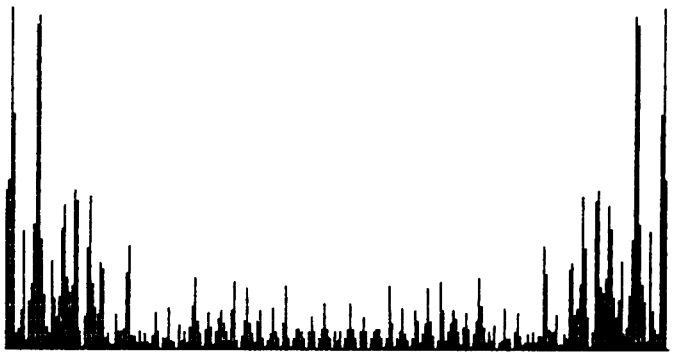
nombre d'echantillons = 256

```
60 FOR i=1 TO n:xr(i)=SGN(SIN(2*PI*i*2.5/N)+SIN(2*PI*i*12.
5/n)):xi(i)=0:NEXT
```

ASPECT TEMPOREL



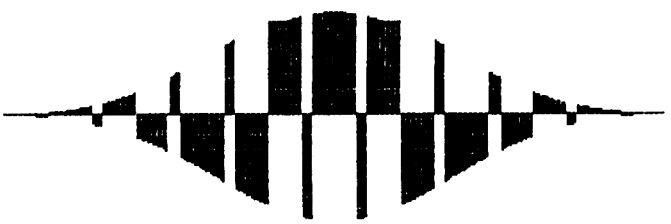
aspect frequentiel



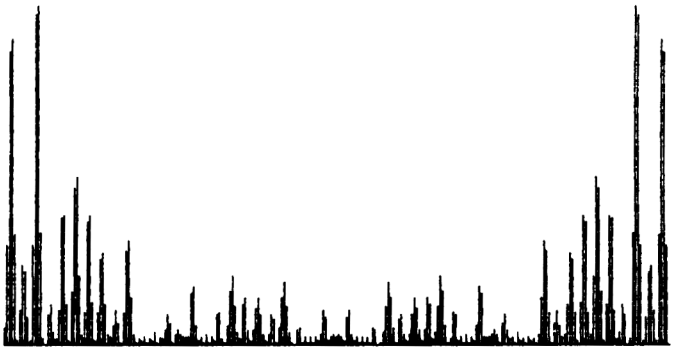
nombre d'echantillons = 256

FENETRE DE BLACKMAN

ASPECT TEMPOREL



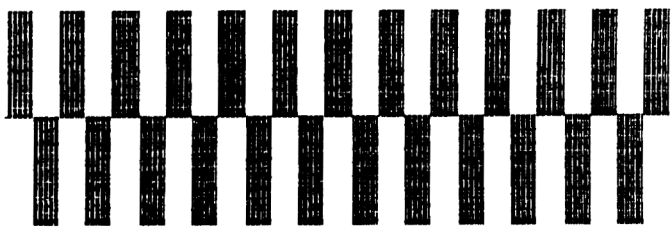
aspect frequentiel



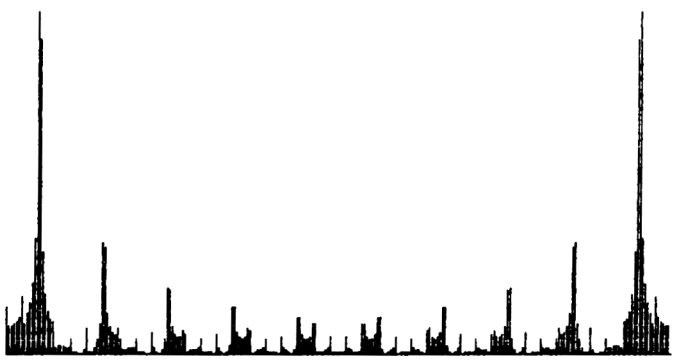
```
60 FOR i=0 TO n:xr(i)=SGN(SIN(2*PI*i*12.5/n)):xi(i)=0:NEXT
```

nombre d'echantillons = 256

ASPECT TEMPOREL



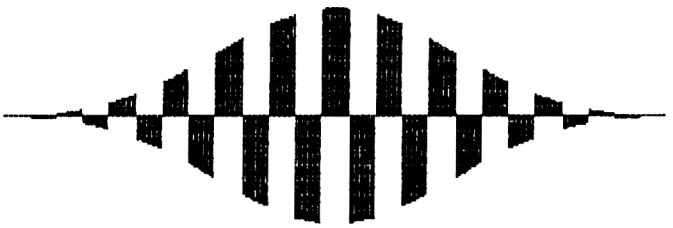
aspect frequentiel



nombre d'echantillons = 256

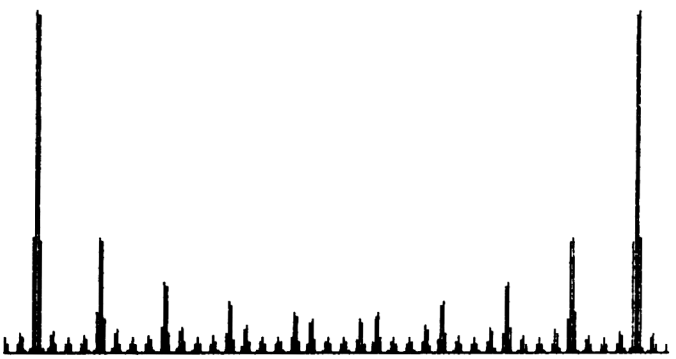
FENETRE DE BLACKMAN

ASPECT TEMPOREL



FENETRE DE BLACKMAN

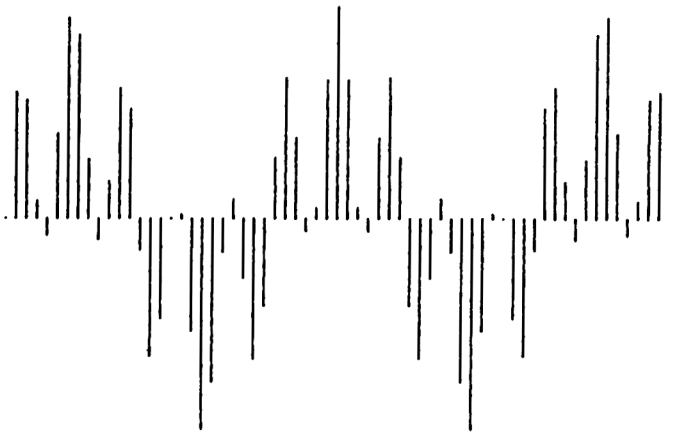
aspect frequentiel



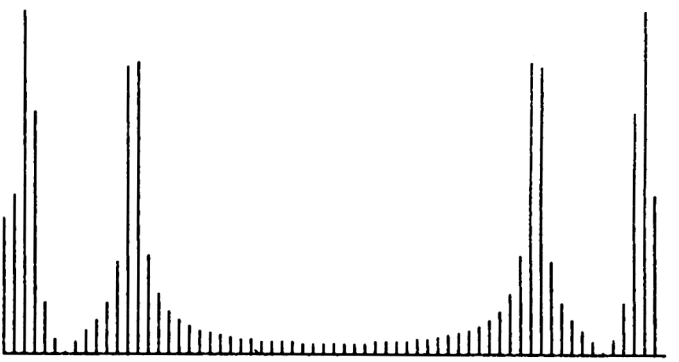
```
60 FOR i=0 TO n:xr(i)=SIN(2*PI*i*12.5/n)+SIN(2*PI*i*2.5/n)
:xi(i)=0:NEXT
```

nombre d'echantillons = 64

ASPECT TEMPOREL



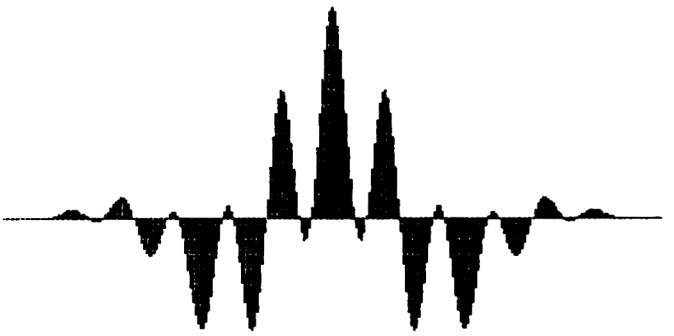
aspect frequentiel



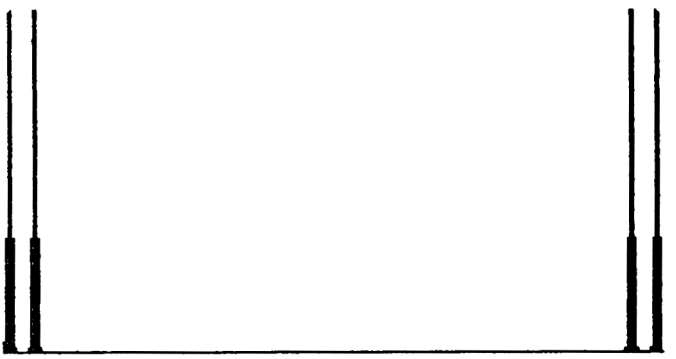
nombre d'echantillons = 256

FENETRE DE BLACKMAN

ASPECT TEMPOREL



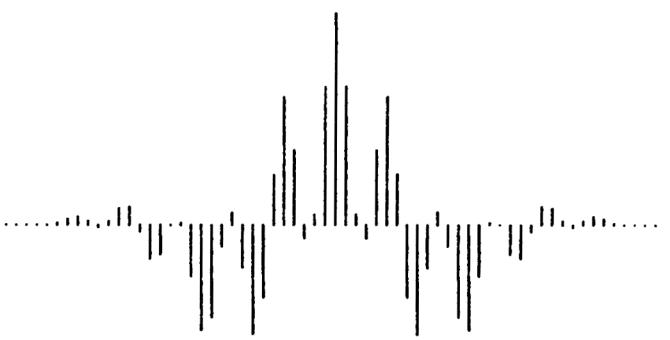
aspect frequentiel



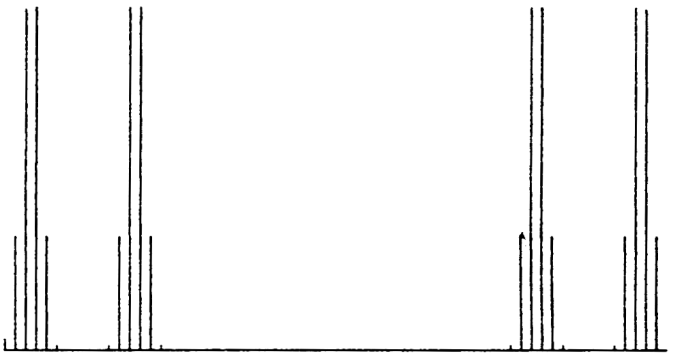
nombre d'echantillons = 64

FENETRE DE BLACKMAN

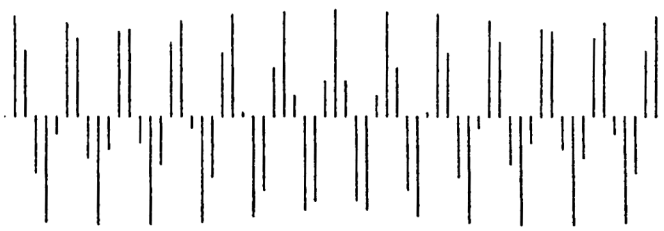
ASPECT TEMPOREL



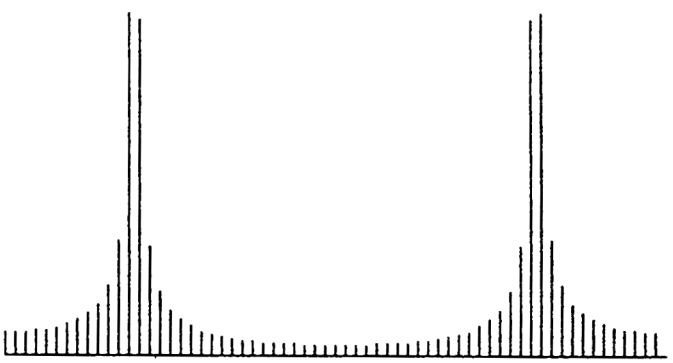
aspect frequentiel



```
60 FOR i=0 TO n:xr(i)=SIN(2*PI*i*12.5/n):xi(i)=0:NEXT
nombre d'echantillons =  64
ASPECT TEMPOREL
```



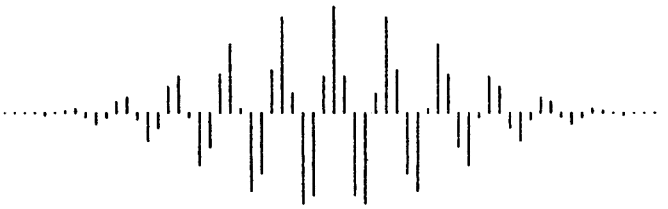
aspect frequentiel



nombre d'echantillons = 64

FENETRE DE BLACKMAN

ASPECT TEMPOREL



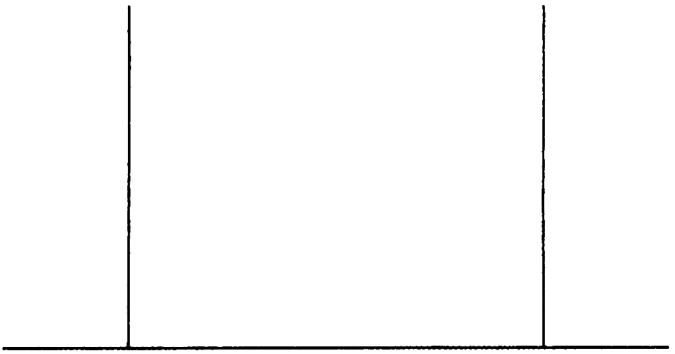
aspect frequentiel

```
60 FOR i=0 TO n:xr(i)=SIN(2*PI*i*12/n):xi(i)=0:NEXT
```

nombre d'echantillons = 64

ASPECT TEMPOREL

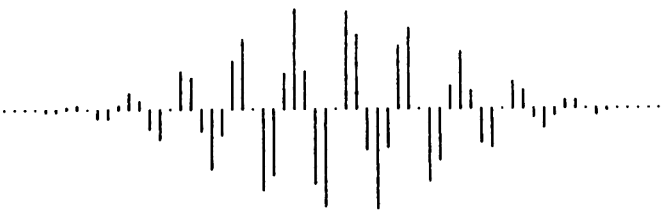
aspect frequentiel



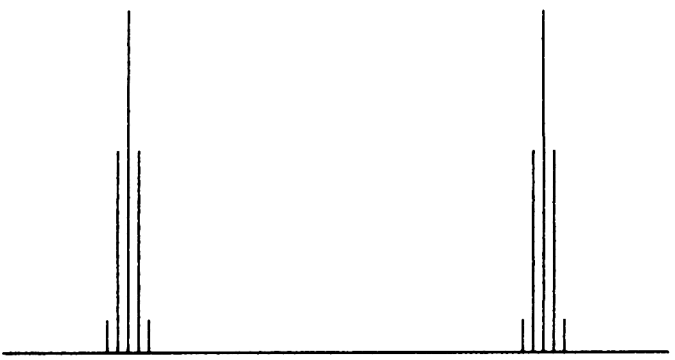
nombre d'echantillons = 64

FENETRE DE BLACKMAN

ASPECT TEMPOREL

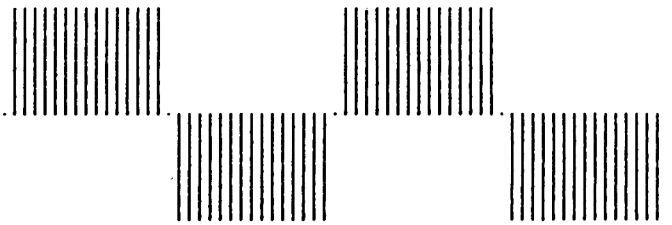


aspect frequentiel

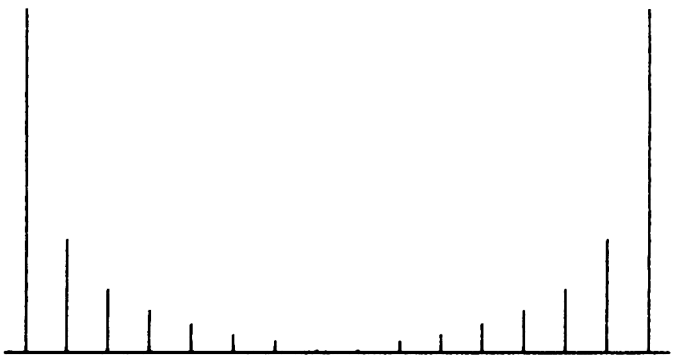


nombre d'echantillons = 64

ASPECT TEMPOREL



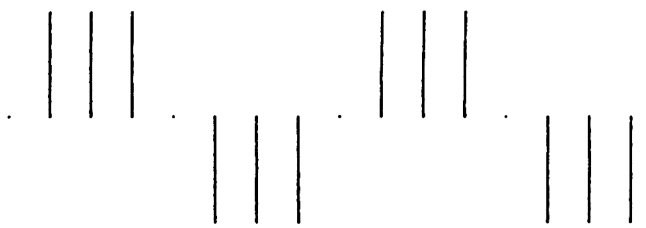
aspect frequentiel



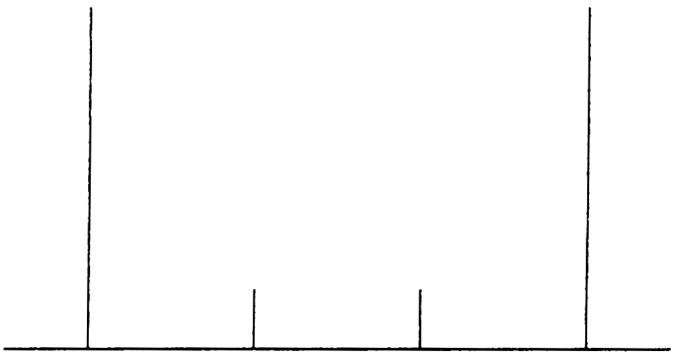
```
60 FOR i=0 TO n:xr(i)=SGN(SIN(2*PI*i*2/n)):xi(i)=0:NEXT
```

nombre d'echantillons = 16

ASPECT TEMPOREL

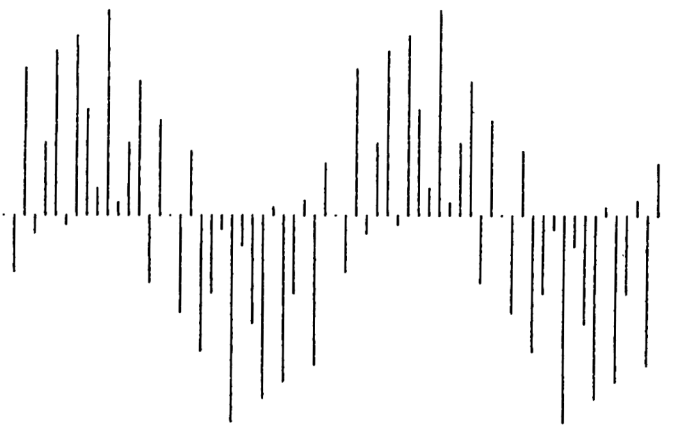


aspect frequentiel

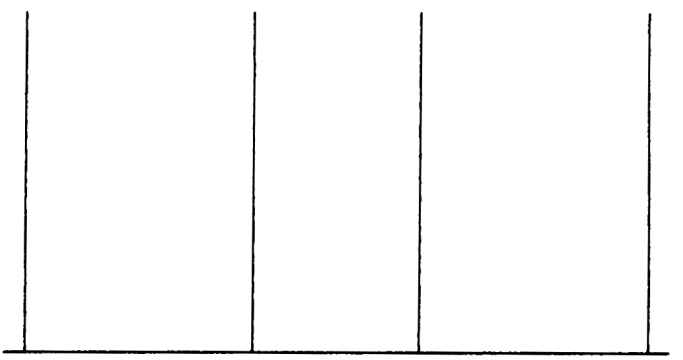


nombre d'echantillons = 64

ASPECT TEMPOREL



aspect frequentiel



```
10 REM *******
20 REM *     *
30 REM * fft *
40 REM *     *
50 REM *******
60 REM
70 REM fft
80 INPUT "nombre d'echantillons:2^n=";n
90 DIM xr(n):DIM xi(n)
100 INPUT"puissance de 2:";nu
110 n2=n/2
120 FOR i=0 TO n:xr(i)=SIN(2*PI*i*2/n):xi(i)=0:NEXT
130 PRINT "fenetre de blackman ? (o/n)"
140 INPUT c$
150 IF c$="n" THEN  200
160 FOR i=0 TO n
170 q=2*PI*(i-n/2)/n
180 xr(i)=xr(i)*(0.42+0.5*COS(q)+0.08*COS(2*q))
190 NEXT i
200 MODE 2
210 FOR I=0 TO N-1
220 PLOT I*512/N,200 +100*XR(I)
230 DRAW I*512/N,200
240 NEXT
250 n1=nu-1
260 k=0
270 FOR l=1 TO nu
280 FOR i=1 TO n2
290 a=INT(k/(2^n1))
300 b=nu
310 GOSUB 680
320 p=br
330 arg=2*PI*p/n
340 c= COS(arg)
350 s=SIN(arg)
360 tr= xr(k + n2)*c+xi(k+n2)*s
370 ti=xi(k+n2)*c-xr(k+n2)*s
380 xr(k+n2)=xr(k)-tr
390 xi(k+n2)=xi(k)-ti
400 xr(k)=xr(k)+tr
410 xi(k)=xi(k)+ti
420 k=k+1
430 NEXT i
440 k=k+n2
450 IF k<n-1 THEN GOTO 280
460 k=0
470 n1=n1-1
480 n2 =INT(n2/2)
490 NEXT l
500 FOR k=0 TO n-1
510 a=k:b=nu
520 GOSUB 680
```

```
530 i=br
540 IF i>k THEN tr=xr(k):ti=xi(k):xr(k)=xr(i)
550 IF i>k THEN xi(k)=xi(i):xr(i)=tr:xi(i)=ti
560 NEXT K
570 MODE 2:CLS
580 FOR i=0 TO n-1
590 xr(i)=SQR(xr(i)^2+xi(I)^2)
600 IF xr(i) > MX THEN MX=xr(i)
610 NEXT
620 PLOT 0,1:DRAW 512,1
630 FOR x=0 TO n-1
640 PLOT x*512/n,0
650 DRAW x*512/n,320*xr(x)/MX
660 NEXT
670 STOP
680 j1=a:z=0
690 FOR x=1 TO b
700 j2=INT(j1/2)
710 z=z*2+(j1-j2*2)
720 j1=j2
730 NEXT x
740 br=z
750 RETURN
```

Nombres aléatoires

Beaucoup d'applications informatiques nécessitent l'emploi de nombres aléatoires, c'est-à-dire fournis par le hasard : jeux, simulations, ... Tous les calculateurs, gros ou petits, disposent donc d'un générateur de tels nombres.

Malheureusement, ces générateurs ne sont pas parfaits et ne peuvent l'être. Ils sont obligés d'utiliser un algorithme, c'est-à-dire une méthode mathématique, fournissant une suite de tels nombres. Ceux-ci ne sont donc pas réellement fournis par le hasard, mais obéissent à une loi : si celle-ci a été bien choisie, ils seront apparemment aléatoires (d'où l'appellation de « pseudo aléatoires »).

L'Amstrad possède un tel générateur ; c'est la fonction RND (RANDOM), qui peut posséder un argument.

– RND (X) avec $X > 0$ ou RND fournit une suite de nombres, aléatoires compris entre 0 (inclus) et 1 (exclus). Après un RUN, la suite fournie débute toujours par le même nombre (0,140522114). C'est l'utilisation classique de cette fonction.

– RND (0) fournit le dernier nombre généré par RND.

– RND (X) avec $X < 0$ génère une nouvelle séquence aléatoire et fournit le premier nombre de cette série.

Le programme 1 montre à l'aide d'un exemple le fonctionnement de ces instructions.

En pratique, on peut avoir besoin d'une série de nombres aléatoires sur un intervalle différent de 0,1. (La simulation d'un jet de dés nécessite des nombres entiers compris entre 1 et 6). Pour cela, la transformation des nombres e fournis par RND se modifie aisément : soit a la borne inférieure de l'intervalle recherché et b sa borne supérieure. Alors :

$r = e\,(b - a) + a$ est un nombre aléatoire compris entre a (inclus) et b (exclus).

r = INT (RND * ($b - a + 1$)) + a est un nombre aléatoire entier (si a l'est) compris entre a et b (bornes incluses).

Ainsi, pour un jeu de dés : ($a = 1, b = 6$) : r = INT (RND * 6) + 1.

L'exemple 1 montre ainsi la génération d'une suite réalisée avec RND, fournissant des nombres compris entre 1 et 10. La première colonne indique le numéro du tirage, la seconde le nombre correspondant et la troisième la moyenne des tirages. Celle-ci devrait valoir 5,5 soit $\frac{1 + 10}{2}$.

Programme 1

```
10 FOR N=1 TO 10
20 PRINT RND(1)
30 NEXT
35 PRINT
40 FOR N=1 TO 10
50 PRINT RND(0)
60 NEXT
65 PRINT
70 FOR N=1 TO 10
80 PRINT RND(-1)
90 NEXT
95 PRINT
100 FOR N= 1 TO 10
110 PRINT RND(1)
120 NEXT
```

```
 0.140522114
 3.67999E-02
 0.182864718
 0.352758596
 0.375612436
 0.342460861
 0.335665084
 0.525869688
 0.595680852
 0.630562254

 0.630562254
 0.630562254
 0.630562254
 0.630562254
 0.630562254
 0.630562254
 0.630562254
 0.630562254
 0.630562254
```

```
 0.271940658
 0.271940658
 0.271940658
 0.271940658
 0.271940658
 0.271940658
 0.271940658
 0.271940658
 0.271940658
 0.271940658

 0.528612386
 0.021330127
 0.175138616
 0.657773343
 0.653729687
 0.601031218
 0.934601486
 0.545995176
 6.34897E-02
 0.989510193
```

```
borne inferieure :  1
borne superieure : 10
 10          9.59671556   5.51875454
 20          5.60180762   5.93379868
 30          6.89482744   6.01776565
 40          1.62745654   5.8957522
 50          5.18088125   5.6442027
 60          9.43551282   5.59505253
 70          9.44405828   5.81809795
 80          2.09893039   5.70822929
 90          5.69826286   5.791301
 100         7.10392081   5.84387646
 110         7.30624957   5.73124676
 120         6.10945942   5.74786858
 130         2.5980467    5.73884861
 140         9.99324489   5.79024574
 150         9.16415391   5.81576419
 160         8.49888545   5.81950822
 170         2.88825701   5.83954354
 180         1.91401477   5.8139789
 190         3.01203209   5.77066382
 200         8.3411812    5.77347873
 210         3.12095485   5.73553977
 220         2.16081473   5.69544628
 230         4.02596864   5.68761453
 240         3.54755386   5.67116233
 250         6.61330752   5.68473864
 260         5.49259977   5.67561483
 270         3.07946628   5.66488768
 280         2.79196698   5.70197194
 290         1.4011865    5.67664178
 300         6.87308257   5.69668658
 310         8.7178112    5.7321342
 320         7.29182588   5.76000798
 330         2.45056281   5.7444562
 340         5.72065724   5.74593601
 350         5.62192494   5.75057956
 360         7.33382604   5.73330317
 370         2.24127767   5.74710013
 380         1.28342976   5.68824169
```

Le programme 2 donne un exemple de générateur de nombres aléatoires, *s* possédant les mêmes propriétés que RND. L'algorithme figure en lignes 100 à 120. Il nécessite une source *s* (initialisation de la série). L'exemple 2 montre l'utilisation de ce programme.

Programme 2

```
10 REM ***********************************
20 REM *                                 *
30 REM * DISTRIBUTION UNIFORME SUR A,B *
40 REM *                                 *
50 REM ***********************************
60 REM
70 INPUT "borne inferieure : ";a
80 INPUT "borne superieure :";b
90 INPUT " source :";s
100 e=(s*24298+99991)/199017
110 e=e-INT(e)
120 s=e*199017
130 r=e*(b-a)+a
140 PRINT r ,
145 m=m+r:n=n+1
146 PRINT m/n
150 GOTO 100
alea1 1
```

```
borne inferieure :  1
borne superieure : 10
 source : 1
 10          5.52856446   6.0516226
 20          4.7407074    6.18213807
 30          8.78338432   6.02858641
 40          4.53052521   6.05688741
 50          5.68249321   5.81847227
 60          9.29811096   5.71420415
 70          7.13658524   5.73042064
 80          3.38451386   5.78859277
 90          6.68521881   5.78629661
 100         7.24133301   5.76884745
 110         8.7949028    5.87846463
 120         3.53142166   5.7867356
 130         3.51099396   5.78708584
 140         6.11974907   5.81517839
 150         8.63453674   5.75193745
 160         2.44633454   5.67338315
 170         4.48486328   5.65779034
 180         2.23352432   5.62686892
 190         5.27956963   5.62435318
 200         8.07586479   5.60518643
 210         4.14936829   5.60236002
 220         2.81775665   5.58374584
 230         1.07093048   5.57088125
 240         7.57198715   5.58641839
 250         7.74557114   5.56818889
 260         8.94035149   5.56048957
 270         2.81104469   5.50233588
```

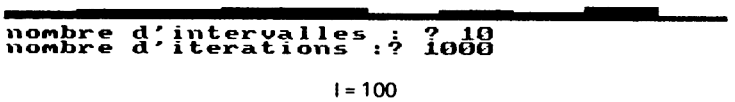
280	9.69869995	5.54281978
290	7.19795418	5.52281415
300	1.01275444	5.51491738
310	1.92147827	5.53204687
320	2.71199608	5.50612589
330	4.00048637	5.4867141
340	5.73672104	5.4604528
350	2.59988403	5.46597608
360	7.91637421	5.48338857
370	3.68028641	5.50259608
380	1.64521074	5.47165818

On peut visualiser la qualité de tels générateurs à l'aide du programme 3. L'intervalle (0,1) est découpé en M intervalles égaux et après chaque tirage, le nombre fourni est placé dans l'intervalle auquel il appartient. Au bout d'un certain temps, l'aspect de l'enveloppe de la courbe devrait être à peu près horizontal, sans trous ni bosses importants. On dit alors que l'on a affaire à une distribution uniforme.

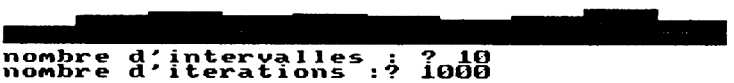
Programme 3

```
1 REM ****************************************
2 REM *                                      *
3 REM * ETUDE DE LA DISTRIBUTION UNIFORME *
4 REM *                                      *
5 REM ****************************************
6 REM
10 CLS
20 REM distribution uniforme
30 INPUT"nombre d'intervalles : ";M
40 INPUT "nombre d'iterations :";IC
50 DIM a(m)
60 FOR i= 1 TO IC
70 sn=RND(1)
80 su=su+sn
90 av=1
100 mn=mn+sn/av
110 ss=ss+sn*sn
120 va=va+sn*sn/(av*av)
130 sn=(sn/av)*m
140 FOR j=0 TO m-1
150 IF INT(sn)= j THEN 180
160 NEXT j
170 GOTO 260
180 a(j)=a(j)+1
```

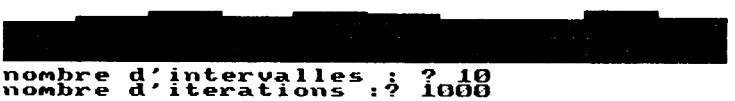
```
190 IF a(j) > 150 THEN 170
200 icr=INT(640/m)
210 PLOT j*icr,a(j):DRAWR icr,0
220 NEXT i
230 i=i-1
240 mn=mn/i:sd=SQR((va-mn*mn/i)/i)
250 PRINT"MOYENNE =";
260 PRINT USING "#.###";mn
270 PRINT "DEVIATION STANDARD =";
280 PRINT USING"#.###";sd
```



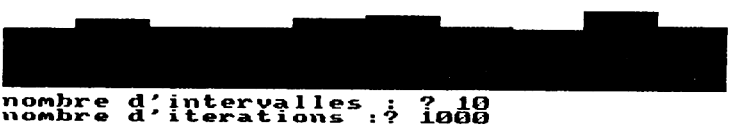
I = 100



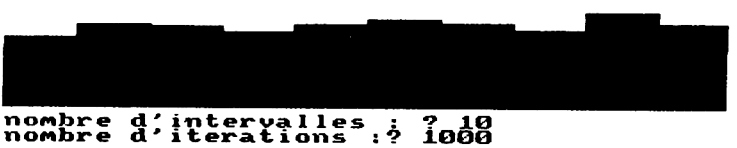
I = 300



I = 500



I = 700



I = 900

Cependant, il n'existe pas que ce type de distribution dans la nature. Si la distribution normale convient pour les jeux de hasard (dés, loto, ...), elle ne s'applique pas à d'autres classes de phénomènes. Ainsi, la distribution des tailles d'individus dans une population donnée, si elle dépend du hasard également, n'est pas pour autant uniforme : si en moyenne elle vaut 1,75 m, il y a peu de chances de trouver des individus de 2,50 m en 0,50 m..

De telles distributions sont dites normales et sont caractérisées par la valeur moyenne *mn* de la variable x_i et par l'écart type (en déviation standard) *sd.*

$$mn = \frac{1}{n}(x_1 + x_2 + \ldots + x_n)$$

$$sd = \text{SQR}\left((n_1 - mn)^2 + (x_2 - mn)^2 + \ldots + (x_n - mn)^2\right)$$

sd renseigne sur la dispersion de la variable autour de la valeur moyenne *mn.*

Il est remarquable que l'on puisse générer de telles suites à partir d'une distribution uniforme. Nous le verrons plus loin.

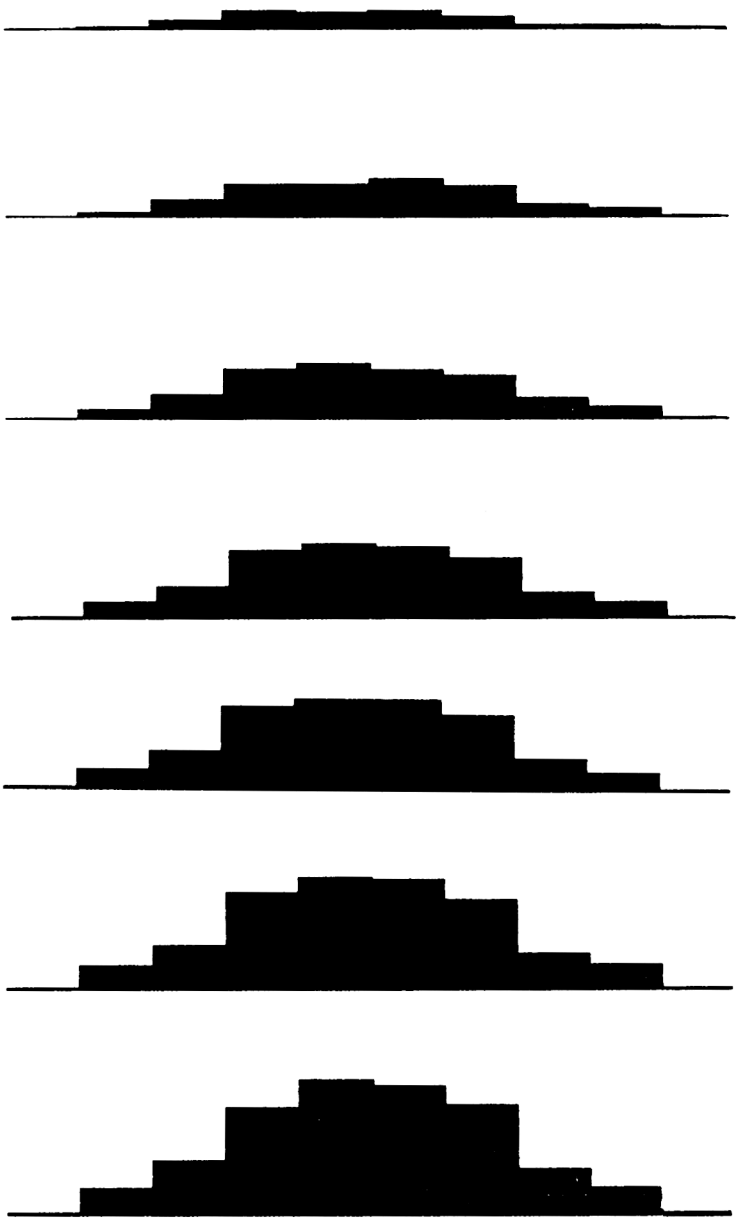
Quel peut être l'intérêt pour un programmeur d'avoir à sa disposition une distribution normale ? Il peut être le suivant. Considérons un mobile qui doit se déplacer sur l'écran du moniteur, dans une direction déterminée. En utilisant des petits déplacements obéissant à la loi normale, ce mobile peut donner l'impression de se déplacer de façon aléatoire, alors qu'en fait il le fera — sur une longue distance — dans la direction choisie.

Revenons à ce type de distribution et tentons d'en réaliser une grâce à la fonction RND. Pour cela, prenons *av* nombres aléatoires successifs dont nous faisons la moyenne *mn*, puis de nouveau les *av* nombres aléatoires suivants, dont nous faisons également la moyenne *mn* et ainsi de suite. Nous disposons ainsi d'une nouvelle suite de nombres qui n'obéissent plus à une distribution uniforme. En effet, chacun d'entre eux n'est plus capable de prendre n'importe quelle valeur comprise entre 0 et 1, mais plutôt une valeur proche de 0,5, valeur vers laquelle tend *mn* si *av* est très grand.

Nous créons ainsi une distribution dite « normale », dont la valeur moyenne est 0,5 et dont l'écart type est réglable par *av*. Le programme 4 permet d'examiner dans le temps l'évolution d'une population régie par une telle loi, (voir l'exemple 4).

Programme 4

```
1 REM ************************
2 REM *                      *
3 REM * DISTRIBUTION NORMALE *
4 REM *                      *
5 REM ************************
6 REM
10 CLS
30 INPUT"nombre d'intervalles : ";M
40 INPUT "nombre d'iterations :";IC
50 INPUT " moyenne effectuee sur : ?";av
60 DIM a(m)
70 FOR i= 1 TO IC
80 sn=0
90 FOR t= 1 TO av
100 sn=sn+RND(1)
110 NEXT t
120 su=su+sn
130 mn=mn+sn/av
140 ss=ss+sn*sn
150 va=va+sn*sn/(av*av)
160 sn=(sn/av)*m
170 FOR j=0 TO m-1
180 IF INT(sn)= j THEN 210
190 NEXT j
200 GOTO 270
210 a(j)=a(j)+1
220 IF a(j) > 150 THEN 200
230 icr=INT(640/m)
240 PLOT j*icr,a(j):DRAWR icr,0
250 NEXT i
260 i=i-1
270 mn=mn/i:sd=SQR((va-mn*mn*i)/i)
280 PRINT"MOYENNE =";
290 PRINT USING "#.###";mn
300 PRINT "DEVIATION STANDARD =";
310 PRINT USING"#.###";sd
```



MOYENNE= 0.49795104

DEVIATION= 0.173601963

nombre d'intervalles : 10

nombre d'iterations : 1000

moyenne effectuee sur : 3

Le programme 5 permet d'étudier la répartition des nombres aléatoires en fonction de *av*. Comme prévu, l'écart type diminue quand *av* augmente.

Il existe d'autres algorithmes permettant d'obtenir une distribution normale à partir d'une distribution uniforme. Par exemple, si n_i et n_{i+1} sont 2 nombres aléatoires successifs (générés par RND), alors la suite :

$$sn = \sqrt{-2.\mathrm{Log}\,(n_i) * \cos\,(2.\pi.n_{i+1})}$$

obéit à une loi normale, de moyenne 0 et d'écart type 1 : la distribution est dite centrée et réduite. De façon plus générale, on peut centrer une distribution normale en soustrayant la valeur moyenne à tous ses éléments et on peut la réduire en divisant toutes les valeurs de la variable par l'écart type.

Inversement, partant d'une distribution centrée réduite (x_i), on peut obtenir une distribution normale (x'_i) de moyenne M et d'écart type ET par la relation :

$$x'_i = x_i\,.\,\mathrm{ET} + \mathrm{M}$$

Enfin, pour ceux que cela intéresse, rappelons que la fonction de densité $f\,(x)$ d'une distribution normale est donnée par la loi :

$$f\,(x_i) = \frac{1}{\sqrt{2\pi}\,\mathrm{ET}}\,.\,\exp\,\left(-\,(x_i - \mathrm{M})^2\,/\,(2\,\mathrm{ET}^2)\right)$$

Programme 5

```
10 REM ****************
20 REM *              *
30 REM * loi normale *
40 REM *              *
50 REM ****************
60 REM
70 INK 0,0:INK 1,26
80 PAPER 0 : PEN 1
90 MODE 2
100 CLS
110 REM distribution normale
120 INPUT"nombre d'intervalles : ";M
130 INPUT "nombre d'iterations :";IC
140 INPUT "moyenne effectuee sur :";av
150 DIM a(m)
160 FOR i= 1 TO IC
170 sn=0
180 FOR t= 1 TO av
```

```
190 sn=sn+RND(1)
200 NEXT t
210 su=su+sn
220 mn=mn+sn/av
230 ss=ss+sn*sn
240 va=va+sn*sn/(av*av)
250 sn=(sn/av)*m
260 FOR j=0 TO m-1
270 IF INT(sn)= j THEN 300
280 NEXT j
290 GOTO 360
300 a(j)=a(j)+1
310 IF a(j) > 150 THEN 290
320 icr=INT(640/m)
330 PLOT j*icr,a(j):DRAWR icr,0
340 NEXT i
350 i=i-1
360 mn=mn/i:sd=SQR((va-mn*mn*i)/i)
370 PRINT"MOYENNE =";
380 PRINT USING "#.###";mn
390 PRINT "DEVIATION STANDARD =";
400 PRINT USING"#.###";sd
410 a=m*su/av
420 b=ss*m*m/av/av
430 ss=SQR((b-A*A/i)/i)
440 s1=sd
450 d=ss*SQR(2*PI)
460 s2=s1*s1
470 REM i/d est le max. de la courbe
480 PLOT 0,i/d:DRAWR 600,0
490 FOR x= 0 TO 1 STEP 1/640
500 num=EXP(-(x-mn)^2/(2*s2))
510 f=x*640
520 g=num*i/d
530 PLOT f,g,,1
540 NEXT
```

nombre d'intervalles : 10

nombre d'iterations : 1000

moyenne effectuee sur : 2

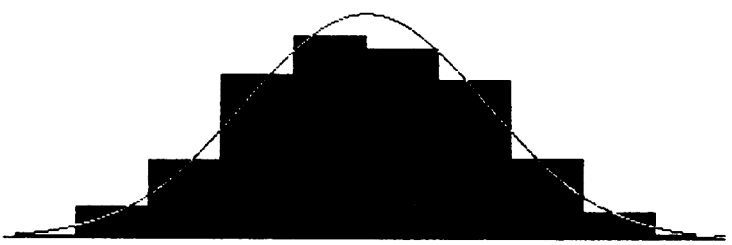
MOYENNE =0.498
DEVIATION STANDARD =0.198

nombre d'intervalles : 10

nombre d'iterations : 1000

moyenne effectuee sur : 3

MOYENNE =0.500
DEVIATION STANDARD =0.166

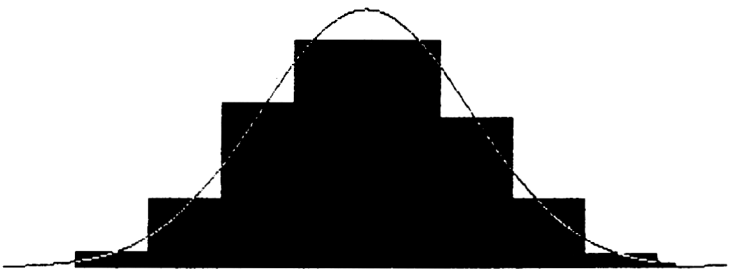


nombre d'intervalles : 10

nombre d'iterations : 1000

moyenne effectuee sur : 4

MOYENNE =0.498
DEVIATION STANDARD =0.145

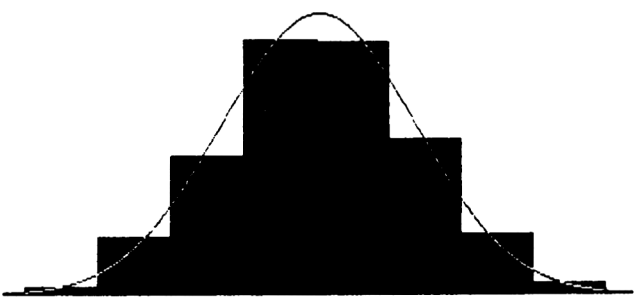


nombre d'intervalles : 10

nombre d'iterations : 1000

moyenne effectuee sur : 5

MOYENNE =0.504
DEVIATION STANDARD =0.134

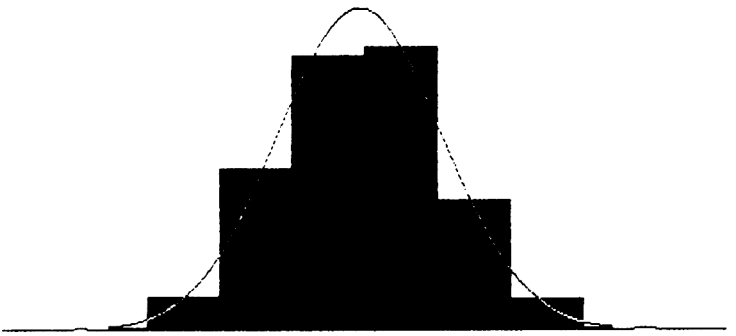


nombre d'intervalles : 10

nombre d'iterations : 1000

moyenne effectuee sur : 6

MOYENNE =0.493
DEVIATION STANDARD =0.114

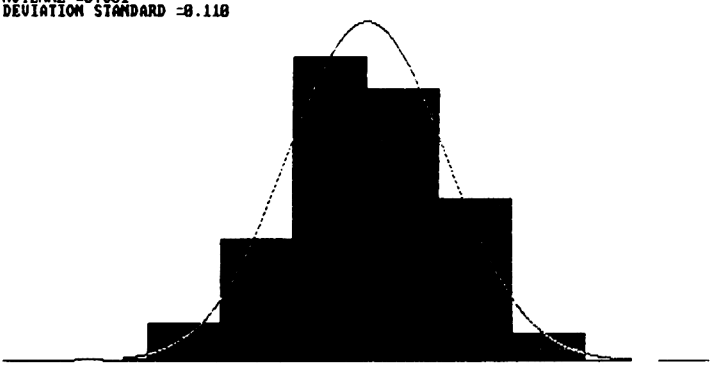


nombre d'intervalles : 10

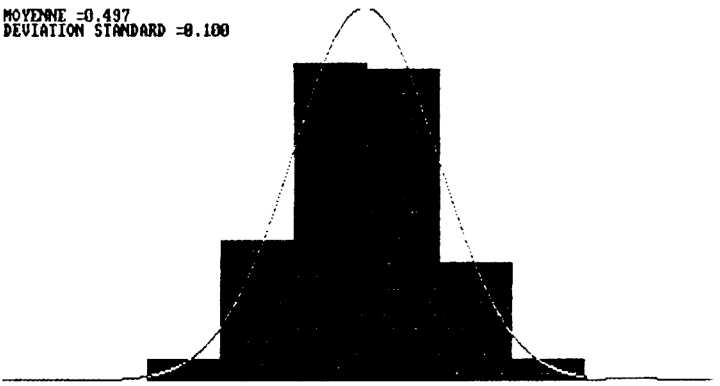
nombre d'iterations : 1000

moyenne effectuee sur : 7

MOYENNE =0.501
DEVIATION STANDARD =0.110



```
nombre d'intervalles :  10

nombre d'iterations : 1000

moyenne effectuee sur : 8
```



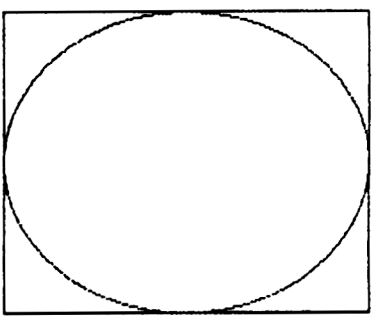
Terminons par une petite application de toutes ces notions : le calcul de π. Considérons un carré de côté a ; sa surface vaut a^2. De même, soit un cercle de rayon $\frac{a}{2}$; sa surface vaut $\pi \frac{a^2}{4}$. Conclusion : le rapport des surfaces vaut $\frac{\pi}{4}$. Plaçons un point au hasard dans le carré ; la probabilité qu'il tombe dans le cercle est évidemment $\frac{\pi}{4}$. On peut donc évaluer π par le rapport :

$$\pi = 4 \times \frac{\text{nombre de points dans le cercle}}{\text{nombre de points dans le carré}}$$

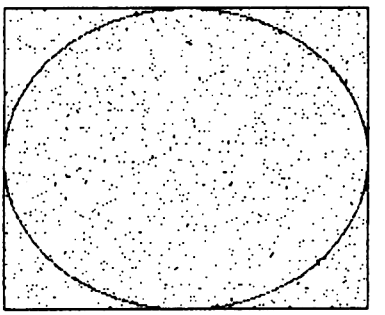
D'où le programme 6 et l'exemple 6.

Programme 6

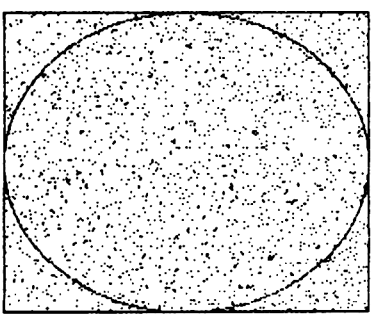
```
10 REM ****************
20 REM *              *
30 REM * calcul de pi *
40 REM *              *
50 REM ****************
60 REM
70 MODE 2
80 PLOT 170,50:DRAWR 0,300
90 DRAWR 300,0:DRAWR 0,-300:DRAWR -300,0
100 FOR n= 0 TO 2*PI STEP PI/300
110 PLOT 320+150*COS(n),200+150*SIN(n)
120 NEXT n
130 FOR i=0 TO 100000
140 x=RND(1)
150 y=RND(1)
160 IF x*x +y*y < 1 THEN p=p+1
170 PLOT 170+x*300,50+y*300
180 IF i/100 <> INT(i/100) THEN GOTO 200
190 LOCATE 1,1: PRINT"pi=";p*4/i
200 LOCATE 1,2:PRINT"i=";i
210 NEXT i
```



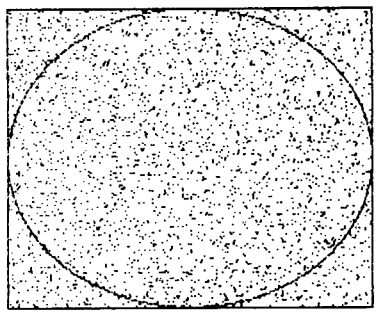
```
i= 100          pi= 3.32

i= 200          pi= 3.1

i= 300          pi= 3.02666667

i= 400          pi= 3.03

i= 500          pi= 3.04

i= 600          pi= 3.05333333

i= 700          pi= 3.02857143

i= 800          pi= 3.015

i= 900          pi= 3.06222222

i= 1000         pi= 3.08

pi= 3.08 07908
i= 1000
```



```
i= 1100         pi= 3.07272727

i= 1200         pi= 3.08

i= 1300         pi= 3.07692308

i= 1400         pi= 3.08571429

i= 1500         pi= 3.09066667

i= 1600         pi= 3.085

i= 1700         pi= 3.08470588

i= 1800         pi= 3.09555556

i= 1900         pi= 3.09684211

i= 2000         pi= 3.1

pi= 3.1 954978
i= 2000
```



```
i= 2100        pi= 3.11428571

i= 2200        pi= 3.11636364

i= 2300        pi= 3.11652174

i= 2400        pi= 3.12333333

i= 2500        pi= 3.136

i= 2600        pi= 3.14923077

i= 2700        pi= 3.15407407

i= 2800        pi= 3.14

i= 2900        pi= 3.13655172

i= 3000        pi= 3.136

pi= 3.136 119
i= 3000
```



Valeur moyenne
Valeur quadratique moyenne

Rappelons tout d'abord les définitions de la valeur moyenne et de la valeur quadratique moyenne d'une fonction $y\ (x)$ sur l'intervalle x_a, x_b :

valeur moyenne : $$\text{MO} = \frac{1}{(x_b - x_a)} \cdot \int_{x_a}^{x_b} y\,(x)\,.\,dx$$

valeur quadratique moyenne : $$\text{QM} = \sqrt{\frac{1}{(x_b - x_a)} \cdot \int_{x_a}^{x_b} (y\,(x))^2\,.\,dx}$$

ce qui implique : QM = valeur moyenne du carré de y.

Dans ce programme l'ordinateur calcule la valeur moyenne et la quadratique moyenne de la fonction $y\ (x)$ proposée par l'utilisateur sur un intervalle de son choix. En outre il représente graphiquement $y\ (x)$ et $(y\ (x))^2$ et met en évidence leur valeur moyenne.

Vous pouvez éviter de recopier les lignes 10 à 140 qui ne servent qu'à la présentation.

Déroulement du programme

L'utilisateur donne la fonction $y\ (x)$ en ligne 1000. Puis il lance le programme en faisant RUN 200. Il indique alors l'intervalle de valeur de x étudié.

L'ordinateur calcule $y\ (x)$ pour 600 valeurs de x dans cet intervalle. Le nombre 600 correspond au nombre de pixels sur la largeur de l'écran (compte tenu des marges). Il calcule une valeur approchée de l'intégrale en faisant la somme des surfaces de 600 rectangles. Il détermine alors la valeur moyenne. Il procède enfin de même avec la fonction $(y\ (x))^2$.

Tout en calculant les valeurs de y, il compare chaque valeur à la précédente afin de connaître les valeurs extrêmales de y sur l'intervalle considéré. Il en déduit une échelle de représentation en ordonnée pour tracer la courbe sur l'écran. Sont également tracées les droites indiquant les valeurs moyennes de y et de y^2.

Après avoir observé y (x) et sa valeur moyenne, on tapera sur la flèche pour obtenir la valeur quadratique moyenne.

Détail du programme

Ligne(s)	
10	préambule
210	intervalle de x considéré
300	boucle de calcul
1000	expression de y (x)
320	mise à jour du maximum et du minimum
340	mise à jour de l'intégrale de y
360	mise à jour de l'intégrale de y
390	calcul des valeurs moyenne et quadratique moyenne
410	détermination du domaine de valeurs de y
510	on étend ce domaine à zéro pour représenter l'axe des abscisses sur l'écran
540	graphe de y (x)
640	graphe de $(y\ (x))^2$

Liste des variables

x	variable
x_a, x_b	bornes du domaine de x
x_c	largeur de ce domaine
ex	échelle en x
dx	accroissement de x entre deux calculs
p, m	valeurs maximale et minimale de y

$$i = \int_{x_a}^{x} y\ (x)\ .\ dx$$

$$j = \int_{x_a}^{x} (y\ (x))^2\ .\ dx$$

$$z = y^2$$

mo valeur moyenne de y
qm valeur quadratique moyenne de y
pp valeur maximale de y^2
ey échelle en ordonnée pour la représentation de y
k décalage en ordonnée pour placer l'axe des abscisses
xx abscisse d'un point sur l'écran

Illustrations

Toutes les illustrations ont été réalisées en prenant pour intervalle de valeur de x : $x_a = 0$, $x_b = 2 . PI$, soit 6,283185.

Comparons les résultats obtenus avec les résultats théoriques :

Figure 1 :

$y = \sin(x)$ [[MO = 0 par raison de symétrie
QO = $1/\sqrt{2}$

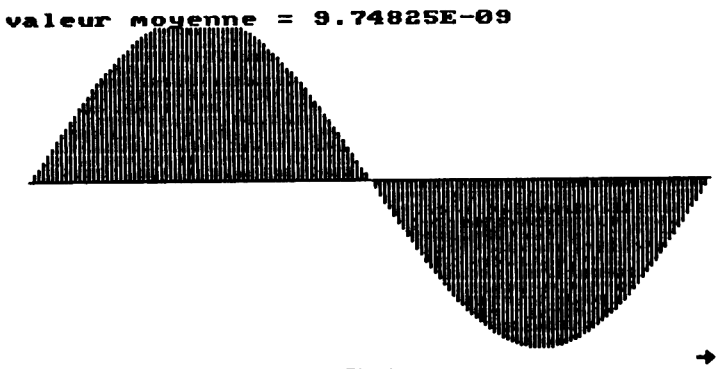


Fig. 1a

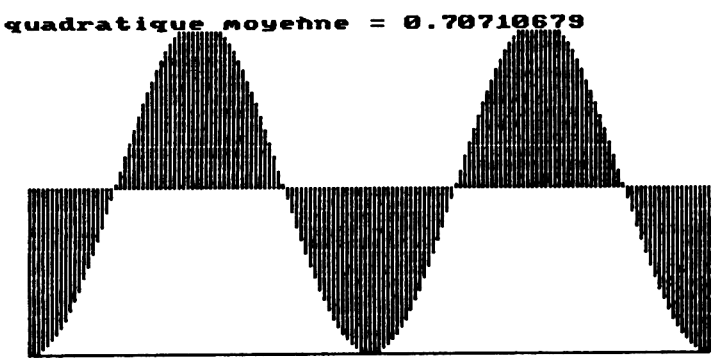


Fig. 1b

Figure 2 :

$y = |\sin(x)|$ [[MO = 2/PI
QM = $1/\sqrt{2}$ (on retrouve la valeur précédente car le signe disparaît dans l'élévation de y au carré).

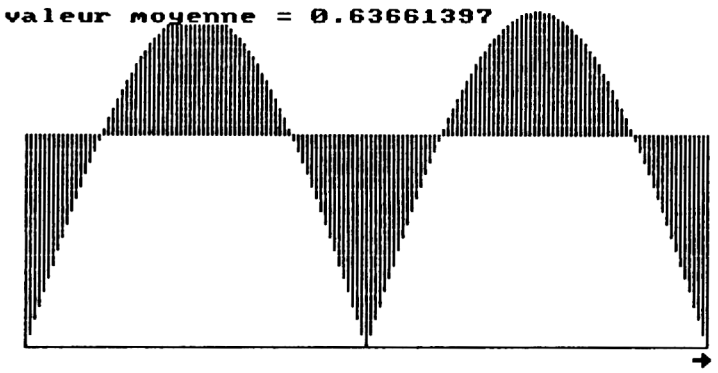


Fig. 2

Figure 3 :

$y = \sin(x)$ si x est inférieur à PI sinon $y = 0$
MO = 1/PI (il n'y a qu'une seule arche, la surface sous la courbe est donc deux fois plus petite).
QM = 1/2 (la surface sous la courbe y est deux fois plus petite donc le carré de la valeur quadratique moyenne est deux fois plus petit).

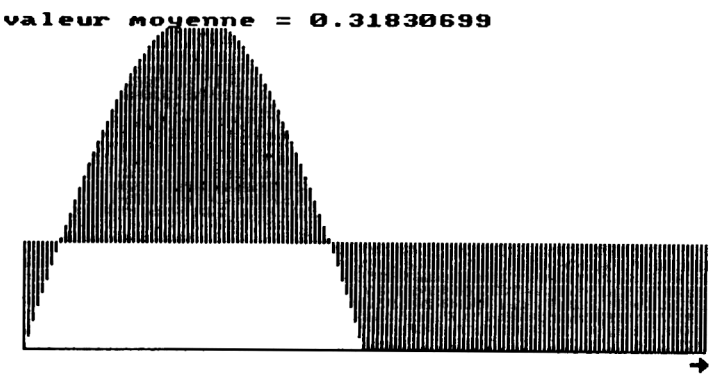
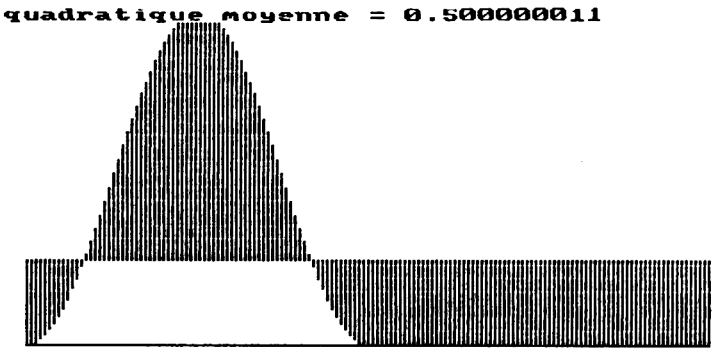


Fig. 3a



3b

Figure 4 :

$y = 2 + \sin(x)$.

MO = 2 (car la valeur moyenne du sinus est nulle).

$y^2 = 4 + 4.\sin(x) + (\sin(x))^2$

- la valeur moyenne de 4 est 4
- la valeur moyenne de $4.\sin(x)$ est nulle
- la valeur quadratique moyenne du sinus est égale à $1/\sqrt{2}$ donc la valeur moyenne de son carré est : $(1/\sqrt{2})^2 = 1/2$.
- finalement QM = $\sqrt{4{,}5}$.

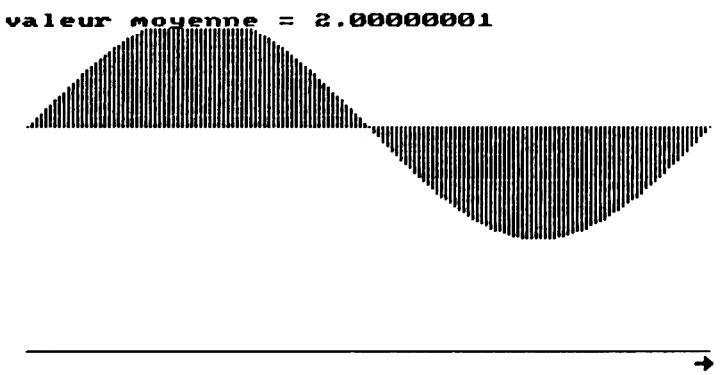
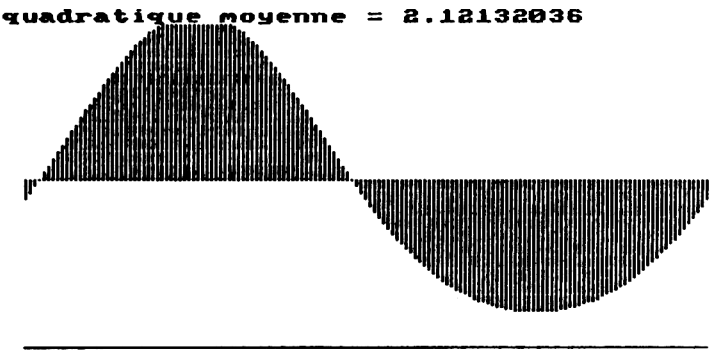


Fig. 4a



4b

Programme

```
10 CLS
20 LOCATE 12,6:PRINT"valeur moyenne"
30 LOCATE 8,8:PRINT"valeur quadratique moyenne"
40 LOCATE 1,22:PRINT"pour faire avancer le program
me, taper "CHR$(243)
50 LOCATE 39,25:PRINT CHR$(243)
60 q=INKEY(1):IF q<>0 THEN 60
70 CLS
100 PRINT:PRINT"donner la fonction y(x)"
110 PRINT:PRINT"en ligne 1000"
120 PRINT:PRINT"puis faire RUN 200"
130 PRINT:PRINT:LIST 1000
140 END
200 CLS
210 PRINT:PRINT"valeur de x initiale "
220 INPUT xa
230 PRINT:PRINT"valeur de x finale "
240 INPUT xb
250 xc=xb-xa
260 ex=600/xc:dx=1/ex
270 PRINT:PRINT"calcul en cours"
280 REM
290 i=0:j=0:x=xa:GOSUB 1000:p=y:m=y
300 FOR x=xa TO xb STEP dx
310 GOSUB 1000
320 IF y>p THEN p=y
330 IF y<m THEN m=y
340 i=i+dx*y
350 z=y*y
360 j=j+dx*z
370 NEXT x
380 REM
390 mo=i/xc
400 qm=SQR(j/xc)
```

```
410 pp=(MAX(ABS(p),ABS(m)))^2
420 REM
500 CLS
510 IF p<0 THEN p=0
520 IF m>0 THEN m=0
530 ey=360/(p-m):k=ey*m
540 MOVE 20,20-k:DRAWR 600,0
550 FOR x=xa TO xb STEP dx*4
560 GOSUB 1000
570 xx=20+(x-xa)*ex
580 MOVE xx,20+mo*ey-k:DRAWR 0,(y-mo)*ey
590 NEXT x
600 PRINT:PRINT"valeur moyenne ="mo
610 LOCATE 39,25:PRINT CHR$(243)
620 q=INKEY(1):IF q<>0 THEN 620
630 CLS
640 ey=360/pp
650 MOVE 20,20:DRAWR 600,0
660 FOR x=xa TO xb STEP dx*4
670 GOSUB 1000
680 MOVE 20+(x-xa)*ex,20+qm*qm*ey
690 DRAWR 0,(y*y-qm*qm)*ey
700 NEXT x
710 PRINT:PRINT"quadratique moyenne ="qm
720 END
1100 RETURN
```

Représentation d'une surface $z\ (x,\ y)$

Après avoir appris à dessiner des courbes, voyons ce qu'il en est des surfaces. Pour bien en concevoir la forme nous allons les représenter en perspective. On peut les suggérer par une distribution de points, mais l'interprétation de cette image est parfois difficile. Aussi nous allons tracer des courbes sur la surface. On peut tracer des courbes à x constant, à y constant, ou à z constant. Cette dernière idée est difficile à mettre en œuvre. Ici nous avons choisi de tracer des courbes à x constant.

Si nous marquons tous les points ou toutes les courbes, l'image est confuse. Aussi nous allons considérer que la surface est opaque et ne marquer que ce qui est vu ; ceci est un problème délicat.

Premier essai

Dans un premier temps, nous regardons la surface par le dessus ($0 < ah < 90°$) et la fonction z est telle que pour un couple x, y il n'y a qu'une seule valeur de z. Vous recopiez pour cela le programme jusqu'en ligne 300 en supprimant toutes les lignes dont le numéro ne se termine pas par zéro.

Le cadrage a été choisi de façon suivante : x et y varient entre $-$ 200 et $+$ 200, z varie entre 0 et 200. Mais il n'est pas interdit d'envisager pour z des valeurs négatives.

La direction du regard est définie par les angles ah et av donnés en degrés suivant la figure 1.

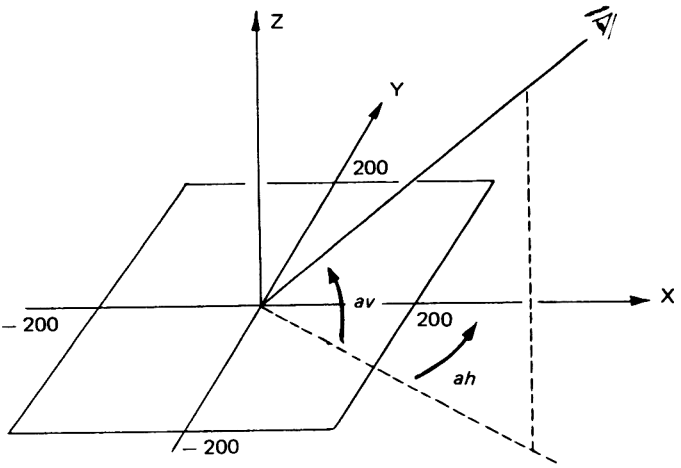


Fig. 1.

Envisageons d'abord la rotation d'angle *ah*. Le fait de ne pas être face au trièdre O*x, y, z* est gênant, repérons-nous donc dans le trièdre O x_1, y_1, z_1. Le point de coordonnées *x, y, z* dans le premier trièdre a maintenant pour coordonnées x_1, y_1, z_1 avec (figure 2) :

$$x_1 = x \,.\, \cos(ah) - y \,.\, \sin(ah)$$
$$y_1 = x \,.\, \sin(ah) + y \,.\, \cos(ah)$$
$$z_1 = z$$

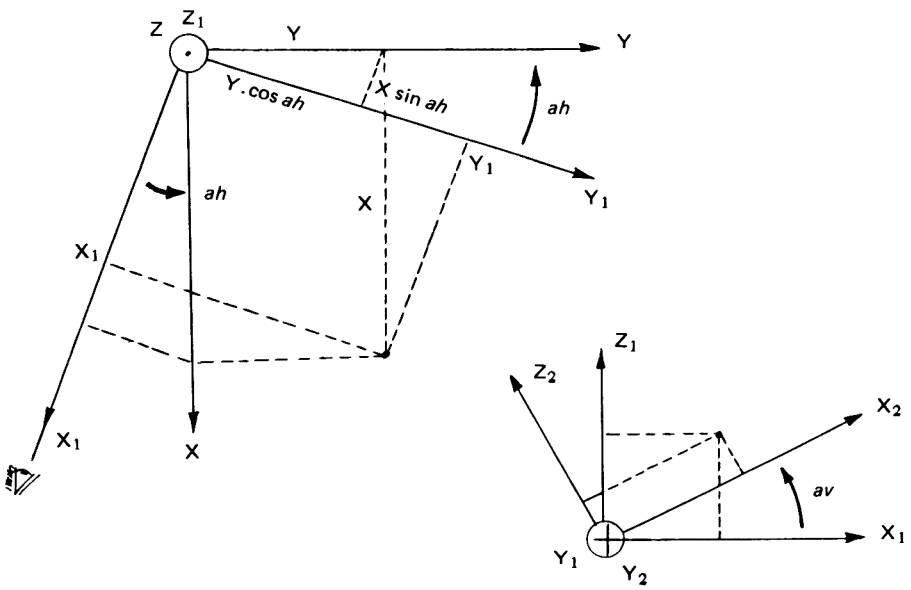


Fig. 2.

Tournons maintenant de *av* et considérons le nouveau trièdr $O\,x_2, y_2, z_2$ avec :

$$x_2 = x_1 \,.\, \cos(av) + z_1 \,.\, \sin(av)$$
$$y_2 = y_1$$
$$z_2 = z_1 \,.\, \cos(av) - x_1 \,.\, \sin(av)$$

Le travail est fait pour chaque point de la surface en ligne 1300 à 1330.

Enfin la surface est vue à travers la fenêtre de l'écran. Les coordonnées du point sur l'écran *xe* et *ye* sont calculées en litne 190 et 200. L'échelle *e* et les décalages 320 et 200 sont à choisir suivant la surface étudiée et l'angle sous lequel on la regarde.

Ne marquer que ce qui est vu

Le volume compris entre $z = 0$ et $z\ (x, y)$ est découpé en tranches. Considérons la première tranche dont l'arête dessine sur l'écran (dans les coordonnées *xe, ye*) une courbe *zz* (*ye*). Dessinons maintenant le faîte de la seconde tranche. A l'abscisse *ye*, le point sur le faîte est à l'ordonnée *ze*, si *ze* est supérieur à *zz* (*ye*) le point est vu donc il est marqué, sinon il ne l'est pas. Si le point est marqué, *ze* est la nouvelle valeur de *zz* (*ye*) (figure 3). Ceci est réalisé en lignes 210 et 230. Les points sont reliés entre eux en ligne 240. Un point étant relié au précédent s'ils sont tous les deux visibles.

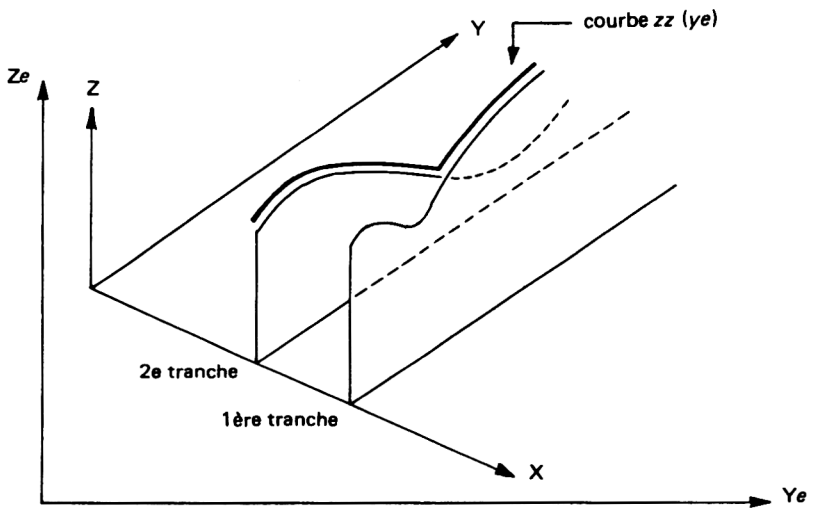


Fig. 3.

Nous pouvons calculer z pour toutes les valeurs entières de x et de y soit 400 × 400 = 60000 points. La haute définition de l'image d'Amstrad serait alors pleinement exploitée mais le programme serait très lent. Aussi nous avons choisi de prendre les valeurs de x de 15 en 15 et les valeurs de ye de 4 en 4 (lignes 190 et 220). Aussi nous avons pris les valeurs de y de 4 en 4 en ligne 160. La haute définition n'est pas inutilisée pour autant puisque les points sont joints par des «droites» de haute définition.

En ligne 40 nous vous proposons un exemple. Avec $e = 1, ah = 30$, $av = 30$, on obtient la figure 4.

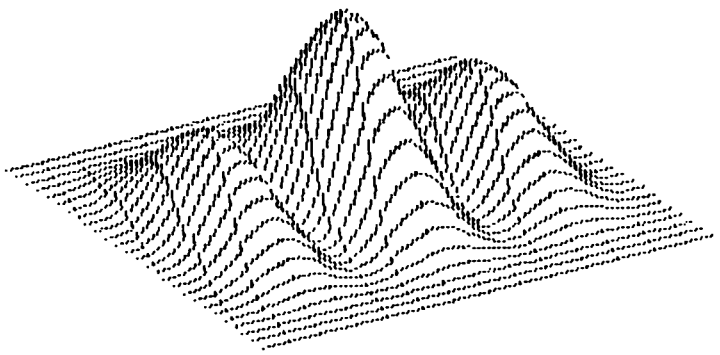


Fig. 4.

Voir le dessous

Si vous essayez des valeurs de av négatives dans le programme précédent, vous constaterez que vous ne pouvez pas voir le dessous de la surface, comme si elle était posée sur un socle opaque. Il faut donc faire un programme parallèle pour traiter le dessous.

Soit $t\ (x, y)$ l'équation du dessous, vous doublez le programme précédent jusqu'en ligne 300 en recopiant les lignes manquantes. C'est lorsque te est supérieur à tt que le point est caché (figure 5).

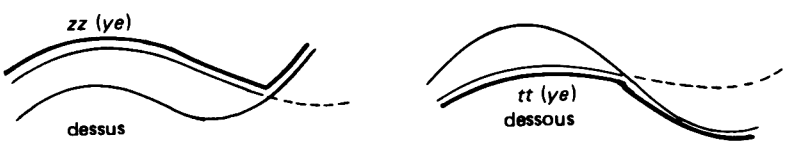


Fig. 5.

Dans un premier temps, regardons le dessous de la surface précédente : en ligne 45 vous placez pour cela : $t = z$. Avec $e = 1$, $ah = 30$, $av = -15$, vous obtenez la figure 6.

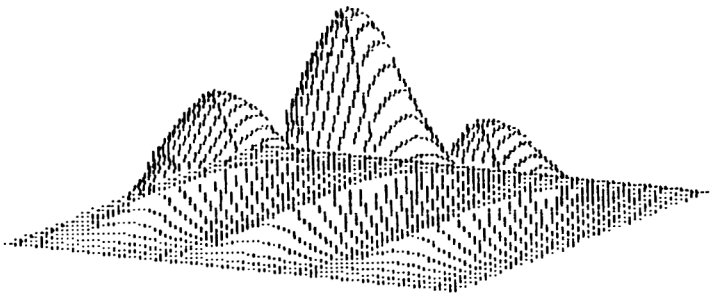


Fig. 6.

Mais vous pouvez aussi représenter un volume en choisissant une surface t différente de la surface z.

Si vous placez en ligne 45 :

$$t = 30 - 10\,000 / (70 + 1{,}3 . \mathrm{MAX}(\mathrm{ABS}(x), \mathrm{ABS}(y)))$$

avec les mêmes angles que précédemment, vous obtenez la figure 7.

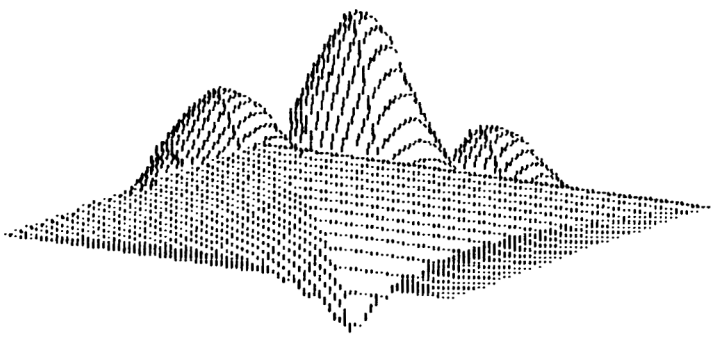


Fig. 7.

Les niveaux

Reprenons la fonction z (x, y) et représentons-là vue de dessus à la manière d'une carte de géographie : plus z est grand, plus la teinte est claire. Pour cela vous recopiez la fin du programme et vous lancez le programme par un RUN 330 si vous ne désirez voir que cette dernière partie.

Le terrain x, y de 400.400 est découpé en rectangles de 8 sur 16 pixels c'est-à-dire la dimension d'un caractère en mode 2. La valeur de z sur ce petit rectangle est comparée à une échelle de 17 valeurs corres-

pondant à 17 caractères plus ou moins pleins. On aurait pu définir de nouveaux caractères avec une répartition de points au hasard dans le petit rectangle et une échelle de 8.16 = 128 valeurs plutôt que d'utiliser les caractères « tout faits » de l'Amstrad.

On retrouve ainsi en figure 8 les trois bosses de la surface z (x, y).

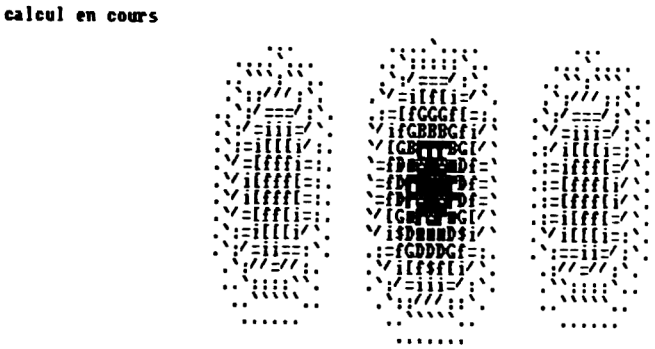


Fig. 8.

Organisation générale du programme

Ligne (s)

40 à 50	définition des surfaces
60 à 130	introduction des angles
140 à 300	dessin des surfaces
170	calcul de z et t
180	envoi à 1300 matrice de rotation
190, 200	coordonnées du point z sur l'écran
210	est-il vu ?
220	il est marqué
240	il est joint au précédent
265 et 262 à 266	même travail pour le point t
330	représentation par niveaux
370	calcul de z
390	liste des caractères
400	adaptation de z à l'échelle des caractères
410 à 440	recherche du caractère
450	affichage

Liste des variables

e échelle
ah, av angles de vue
sh, ch, sv, cv lignes trigonométriques de ces angles

x, y, z, t coordonnées des points « dans la réalité »
si $d = 1$ le point de dessus est vu
si $c = 1$ le point de dessous est vu
$x_1, y_1, z_1, x_2, y_2, z_2$ intermédiaires de calcul
ye, ze, te coordonnées des points sur l'écran
zz (*ye*) valeur maximale précédente de *ze* pour l'abscisse *ye*
tt (*ye*) valeur minimale précédente de *te* pour l'abscisse *ye*
zz, tt valeurs précédentes de *ze* et *te*
w correspond à *z* dans l'échelle des niveaux
ww, n intermédiaires de calcul

Programme

```
10 REM"surface z(x,y)"
20 REM
30 CLS:GOTO 60
40 z=200*((COS(y*0.024))^2)*(COS(y*0.008))*((COS(x
*0.008))^2)*(COS(x*0.008))
45 t=z
50 RETURN
60 DEG
70 PRINT:INPUT"e=";e
80 PRINT:INPUT"ah (-90< <90 )=";ah
90 PRINT:INPUT"av ( 0< <90 )=";av
100 sh=SIN(ah):ch=COS(ah)
110 sv=SIN(av):cv=COS(av)
120 RAD:CLS
130 DIM zz(160)
135 DIM tt(160)
136 FOR n=0 TO 160: tt(n)=400
137 NEXT n
140 FOR x=200 TO -200 STEP -15
150 d=0
155 c=0
160 FOR y=-200 TO 200 STEP 4
170 GOSUB 40
180 GOSUB 1300
190 ye=INT((320+e*y2)/4)
200 ze=200+e*z2
205 te=200+e*t2
210 IF ze<zz(ye) THEN GOTO 260
220 PLOT 4*ye,ze
230 zz(ye)=ze
240 IF d=1 THEN DRAWR ye-yy,ze-zz
250 d=1
260 zz=ze:yy=ye
262 IF te>tt(ye) THEN GOTO 275
264 PLOT 4*ye,te:tt(ye)=te
266 IF c=1 THEN DRAWR ye-yy,te-tt
268 c=1
270 REM
```

```
275 tt=te
280 NEXT y
290 NEXT x
300 REM
310 LOCATE 39,25:PRINT CHR$(243)
320 q=INKEY$(1):IF q<>0 THEN GOTO 320
330 MODE 2:PRINT"calcul en cours"
340 TAG
350 FOR x=-180 TO 180 STEP 16
360 FOR y=-200 TO 200 STEP 8
370 GOSUB 40
380 MOVE 330+y,200-x
390 DATA 32,46,96,58,192,61,105,91,102,36,71,68,66
,233,135,225,143
400 w=INT((z+13)/12)
410 FOR n=1 TO w
420 READ ww
430 NEXT n
450 PRINT CHR$(ww);
455 RESTORE
460 NEXT y
470 NEXT x
480 TAGOFF
500 CALL &A000
510 END
1300 REM
1320 x1=x*ch-y*sh
1325 t1=t
1330 y1=x*sh+y*ch
1340 z1=z
1350 x2=x1*cv+z1*sv
1355 t2=t1*cv-x1*sv
1360 y2=y1
1370 z2=z1*cv-x1*sv
1380 RETURN
```

Représentation d'une surface en coordonnées cylindriques

Nous nous proposons maintenant de représenter une surface définie par une équation en coordonnées cylindriques. Le volume enfermé par la surface est d'abord découpé en tranches (comme une orange), puis découpé en disques.

Compromis

La technique utilisée pour représenter la surface z (x, y) est difficilement utilisable ici. Nous devons nous orienter vers une autre méthode qui nécessite la mise en mémoire de l'état des points de l'écran. Ceci utilisant une mémoire très importante, nous sommes amenés à travailler en basse résolution ce qui procure d'autre part l'avantage d'un travail rapide au détriment de la qualité de l'image. C'est en tout cas un bon compromis pour mettre au point le programme et le cadrage.

En fait nous gâchons de la mémoire car nous stockons dans un tableau une variable binaire : 0 ou 1. Nous n'utiliserions pas plus de mémoire en stockant dans chaque emplacement un nombre de 4 chiffres dans lequel on pourrait lire 4 valeurs de la variable binaire. A vous de compléter le programme dans ce sens afin de profiter pleinement de la haute résolution de l'écran.

Méthode

Tout d'abord, nous considèrerons des équations r $(z; a)$ telles qu'à une valeur de z corresponde une seule valeur de r. Le cadrage est prévu pour un rayon maximal de 99 et une variation de z de 0 à 200.

Recopiez donc le programme suivant en supprimant toutes les lignes dont le numéro ne se termine pas par zéro.

L'équation r (z, a) définit un volume que nous allons découper en tranches, donc à a constant. Nous décrivons les tranches en partant du bas (z = 0) jusqu'en haut (z = 200). Nous commençons par les tranches de devant à gauche, pour aller derrière à gauche puis nous revenons devant à droite pour aller derrière à droite. Le solide est vu en perspective, la rotation ah du regard revenant à faire un changement de variable dans l'équation : la tranche a devient la tranche $aa=a-ah$ (ligne 20).

Pour une valeur de z et de a on en déduit r donc les coordonnées «réelles» x, y, z du point (ligne 250). La rotation av du plan d'observation modifie ces valeurs en $y2$ = y et $z2$ comme dans le programme «z (x, y)» (ligne 270). e est l'échelle de reproduction. $z3$ est l'ordonnée du point sur l'écran, 50 correspond au cadrage et peut être modifié suivant les besoins. Pour un essai, on n'utilise qu'un point sur 16 de l'écran, les coordonnées sont donc comptées de 4 en 4 (lignes 280 et 310), $y4$ et $z4$ sont les indices du point sur l'écran.

A une valeur de z correspond un point du bord de la tranche et un couple $y4$,$z4$ sur l'écran. Ces coordonnées sont mémorisées rp% et zp% (ligne 320). Si nous sommes sur la première tranche, le point est marqué et joint au précédent (ligne 340). En faisant varier z de 0 à 200 nous marquons ainsi tout le bord de la tranche. Ce contour et

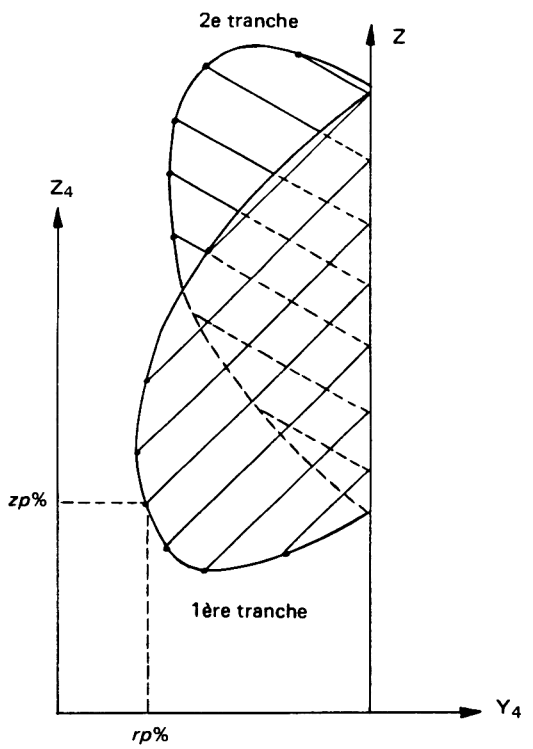


Fig. 1.

l'axe central définissent une surface fermée. En ligne 390 à 450 nous affirmons que cette tranche est opaque et que l'on ne pourra plus marquer de points à l'intérieur de cette surface fermée (figure 1). Dans cette première étude, *rm*% est constamment nul. Pour visualiser ce travail, vous pouvez ajouter la ligne 445 :

PLOT 320 + 4.*w.s*, 4.*t* (figure 2)

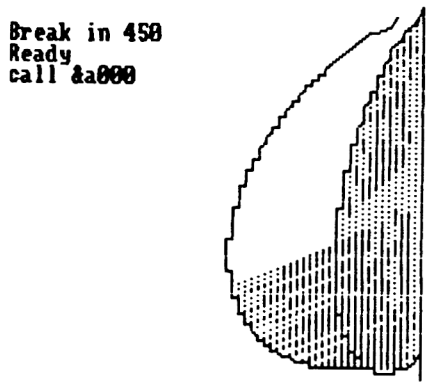


Fig. 2.
L'ordinateur est en train de condamner les points de la seconde tranche.

L'état de chaque point de l'écran (en fait un point sur 16) est noté *p*%(*y*4,*z*4).

Lors du dessin de la tranche suivante on testera la valeur de *p*% avant de marquer un point (ligne 330).

Les points à l'intérieur de la tranche sont répartis sur des rayons perpendiculaires dans la réalité à l'axe du solide, ils sont donc tracés à *z* constant ce qui nécessite un calcul assez pénible en lignes 420 à 450. Nous aurions pu balayer la tranche par des lignes horizontales sur l'écran (donc à *z*4 constant) mais la définition du début et de la fin de ces lignes est complexe.

Le travail est fait pour la partie gauche de la figure ($s = -1$) puis pour la partie droite ($s = 1$).

Illustrations

Après avoir vérifié le bon fonctionnement du programme et le cadrage, vous doublez la résolution avec :

DIM *p*% (50,200) en ligne 130
*z*4 = INT (*zz*/2) en ligne 310

et en remplaçant 4.*z*4 et 4 . (*zz*-*z*4) par 2 . *z*4 et 2 . (*zz*-*z*4) en ligne 340. Vous augmentez également le nombre de tranches en remplaçant 0,4 par 0,2 ou 0,1 en ligne 200.

Nous vous proposons quelques équations en fin de programme, il suffit de les placer en lignes 40 à 49 pour observer la figure correspondante (nettoyez l'espace 40, 49 entre chaque essai). La « tour » nécessite un recadrage ; remplacez 50 par 150 en lignes 160 et 275.

Figure 3 citrouille $e=1,5$ $ah=0$ $av=45$

```
Break in 265
Ready
call &a000
```

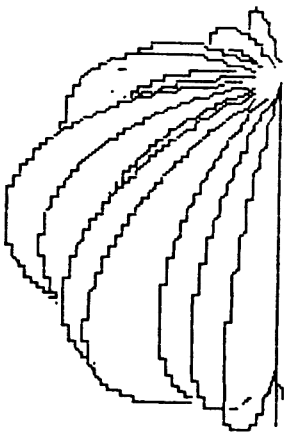


Fig. 3.
La moitié gauche de la citrouille.
($s = -1$)

tour $e=0,8$ $ah=35$ $av=30$

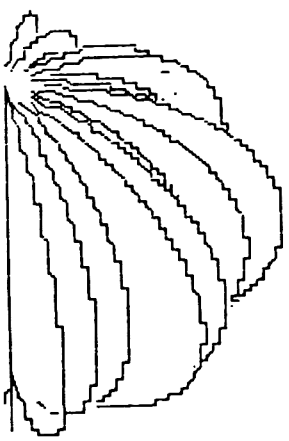


Fig. 4.
La moitié droite de la citrouille.
($s = 1$)

Fig. 4.

Organisation du programme

Ligne(s)

40	équation de la surface
60	introduction des données
130	préparation des tableaux
190	chaque côté
200	chaque tranche
250	où est le point ?
330	est-il vu ?
390	chaque tranche est opaque
490	mise à zéro des *p* %
540	exemples

Liste des variables

r, z, a, x, y coordonnées du point « dans la réalité »
e échelle
ah, av angles de vue
sh, sv, cv lignes trigonométriques de ces angles
aa angle *a* corrigé de *ah*
si *p*%(*y*4,*z*4) = 1, il ne faut plus marquer de point à cet endroit
u (*z*) ordonnée sur l'écran du point de l'axe central d'ordonnée *z*
rp%(*z*) valeur maxmale de l'abscisse sur l'écran pour *z* et *aa*
*z*2,*z*3,*k* intermédiaires de calcul
*y*4,*z*4 coordonnées du point sur l'écran
yy,zz coordonnées du point précédent
si *d* = 1 le point est vu
w,t coordonnées du point écran rendu « opaque »

Premier programme

```
10 REM"surface r(z,a) "
20 REM
30 CLS:GOTO 60
40 r=(1-0.15*SIN(9*a))*SQR(10000-(z-100)^2)

50 RETURN
60 DEG
70 PRINT:INPUT"e=";e
80 PRINT:INPUT"ah (-90< <90 )=";ah
90 PRINT:INPUT"av ( -90< <90 )=";av
100 ah=ah*PI/180
110 sv=SIN(av):cv=COS(av)
```

```
120 CLS:RAD
130 DIM p%(50,100):DIM u(200)
140 DIM rp%(200):DIM zp%(200)
145 DIM rm%(200)
150 FOR z=0 TO 200
160 u(z)=INT((50+e*z*cv)/4)
170 NEXT z
180 REM
190 FOR s=-1 TO 1 STEP 2
200 FOR aa=0 TO s*PI STEP s*0.4
210 a=aa+ah
220 d=0
225 c=0
230 REM
240 FOR z=0 TO 200
250 GOSUB 40
260 x=r*COS(aa):y=r*SIN(aa)
265 xb=r2*COS(aa):yb=r2*SIN(aa)
270 z2=z*cv-x*sv
275 zb=50+e*(z*cv-xb*sv)
280 y4=INT(e*ABS(y)/4)
285 by=INT(e*ABS(yb)/4)
290 z3=50+e*z2
300 IF z3<0 OR z3>400 THEN GOTO 370
310 z4=INT(z3/4)
320 rp%(z)=y4:zp%(z)=z4
330 IF p%(y4,z4)=1 THEN GOTO 360
340 IF d=1 THEN PLOT 320+4*y4*s,4*z4:DRAWR 4*(yy-y
4)*s,4*(zz-z4)
350 d=1
360 yy=y4:zz=z4
362 IF zb<0 OR zb>400 THEN GOTO 370
363 bz=INT(zb/4):rm%(z)=by
364 IF p%(by,bz)=1 THEN GOTO 367
365 IF c=1 THEN PLOT 320+4*by*s,4*bz:DRAWR 4*(yp-b
y)*s,4*(zp-bz)
366 c=1
367 yp=by:zp=bz
370 NEXT z
380 REM
390 FOR z=0 TO 200
400 IF rp%(z)=0 THEN GOTO 460
410 k=(zp%(z)-u(z))/rp%(z)
420 FOR w=rm%(z) TO rp%(z)
430 t=u(z)+w*k
440 p%(w,t)=1
450 NEXT w
460 NEXT z
470 REM
480 NEXT aa
490 FOR w=0 TO 50:FOR t=0 TO 100
500 p%(w,t)=0
510 NEXT t:NEXT w
520 NEXT s
530 END
540 REM "tour"
550 b=100+70*COS(2*a)
560 IF z>b THEN r=80 ELSE r=200
570 REM "champignon"
580 IF z<80 THEN r=30:GOTO 50
```

```
590 IF z>100 THEN r=SQR(10000-(z-100)^2):GOTO 50

600 r=3.5*z-250
610 REM "citrouille"
620 r=(1-0.15*SIN(9*a))*SQR(10000-(z-100)^2)

630 REM "tore"
635 REM "tore"
645 rr=SQR(2500-(z-100)^2)
655 r=140+rr:r2=140-rr
```

Améliorons le programme.

De même que vous avez doublé le programme «*z* (*x*, *y*)» pour observer le dessous, de même en complétant votre programme *r* (*z*, *a*) par les lignes manquantes vous allez pouvoir observer l'intérieur de la surface *r* (*z*, *a*). Vous pourrez envisager alors les équations de surface telles que, pour une valeur de *z* et de *a*, il y ait deux valeurs possibles pour *r*.

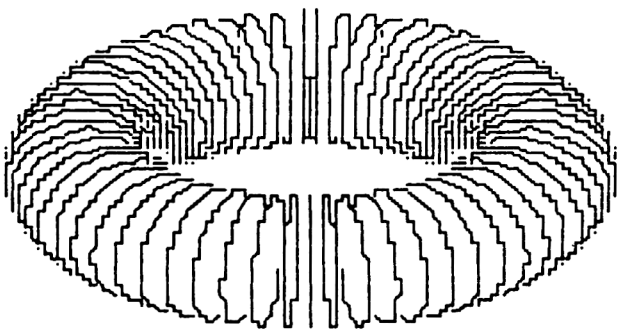


Fig. 5.

En figure 5, l'exemple du tore vous montre l'effet obtenu. La figure 6 précise les notations et voici enfin la correspondance, pas toujours heureuse il est vrai, entre les variables relatives aux deux parties de la surface.

(en ligne 240 vous avez intérêt à faire varier *z* de 50 à 150 seulement.)

EXTERIEUR	*rp*%	*r*	*x*	*y*	*y*4	*z*3	*z*4	*yy*	*zz*
INTERIEUR	*rm*%	*r*2	*xb*	*yb*	*by*	*zb*	*bz*	*yp*	*zp*

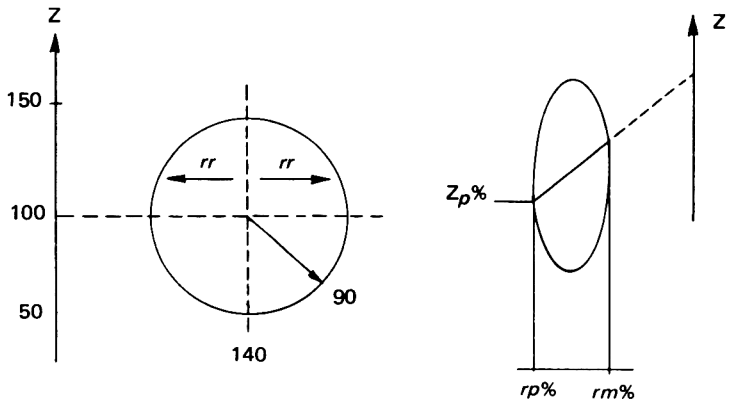


Fig. 6.

Empilement de disques

En permutant les variations de *z* et de *aa*, nous découpons le volume en tranches perpendiculaires à l'axe des *z*. Nous avons préféré vous donner le LIST complet bien qu'il n'y ait que très peu de modifications par rapport au programme précédent. Nous reprenons en figure 7 et 8 les exemples déjà traités.

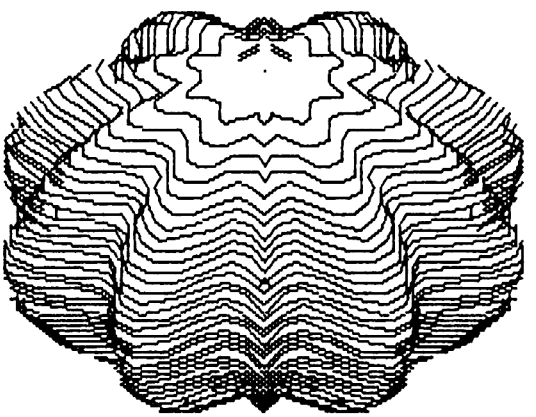


Fig. 7.

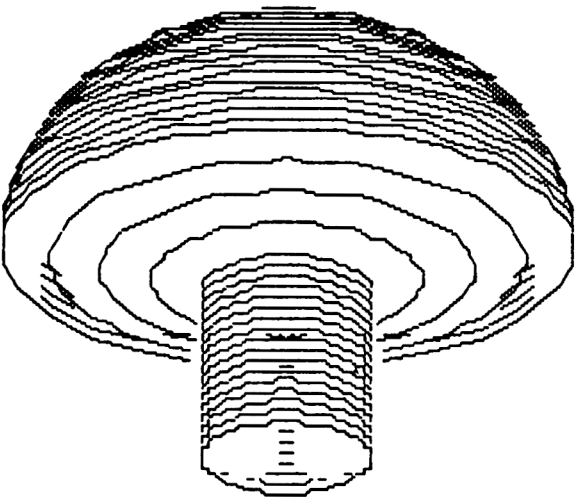


Fig. 8.

Deuxième programme

```
10 REM"surface r(z,a) en disques"
20 REM
30 CLS:GOTO 60
40 r=(1-0.15*SIN(9*a))*SQR(10000-(z-100)^2)

50 RETURN
60 DEG
70 PRINT:INPUT"e=";e
80 PRINT:INPUT"ah (-90< <90 )=";ah
90 PRINT:INPUT"av (0 < <90 )=";av
100 ah=ah*PI/180
105 g=SGN(av):IF g=0 THEN g=1
110 sv=SIN(av):cv=COS(av)
120 CLS:RAD
130 DIM p%(50,100):DIM u(200)
140 DIM rp%(32):DIM zp%(32)
150 FOR z=0 TO 200 STEP 5
160 u(z)=INT((50+e*z*cv)/4)
170 NEXT z
180 REM
190 FOR s=-1 TO 1 STEP 2
195 FOR z=100+g*100 TO 100-g*100 STEP -5*g
200 d=0: FOR n=0 TO 31
210 a=(n/10)+ah
230 REM
250 GOSUB 40
260 x=r*COS(a):y=r*SIN(a)
270 z2=z*cv-x*sv
```

```
280 y4=INT(e*ABS(y)/4)
290 z3=50+e*z2
300 IF z3<0 OR z3>400 THEN GOTO 370
310 z4=INT(z3/4)
320 rp%(n)=y4:zp%(n)=z4
330 IF p%(y4,z4)=1 THEN GOTO 360
340 IF d=1 THEN PLOT 320+4*y4*s,4*z4:DRAWR 4*(yy-y
4)*s,4*(zz-z4)
350 d=1
360 yy=y4:zz=z4
370 NEXT n
380 REM
390 FOR n=0 TO 31
400 IF rp%(n)=0 THEN GOTO 460
410 k=(zp%(n)-u(z))/rp%(n)
420 FOR w=0 TO rp%(n)
430 t=u(z)+w*k
440 p%(w,t)=1
450 NEXT w
460 NEXT n
470 REM
480 NEXT z
490 FOR w=0 TO 50:FOR t=0 TO 100
500 p%(w,t)=0
510 NEXT t:NEXT w
520 NEXT s
530 END
540 REM "tour"
550 b=100+70*COS(2*a)
560 IF z>b THEN r=80 ELSE r=200
570 REM "champignon"
580 IF z<80 THEN r=30:GOTO 50
590 IF z>100 THEN r=SQR(10000-(z-100)^2):GOTO 50

600 r=3.5*z-250
610 REM "citrouille"
620 r=(1-0.15*SIN(9*a))*SQR(10000-(z-100)^2)
```

Représentation «Fil de fer» de polyèdres

Nous allons apprendre sur un exemple à représenter les arêtes de polyèdres convexes vus en perspective en supprimant les arêtes cachées.

L'exemple choisi est un polyèdre à 10 sommets et 9 faces. Chaque face est un polygone plan. Ce polygone a au plus 4 sommets. Après avoir étudié cet exemple, vous pourrez réaliser sans peine des dessins de polyèdres à plus grand nombre de sommets, de faces, et dont les polygones ont un plus grand nombre de sommets.

Définition du polyèdre

En ligne 40 on prépare les tableaux de données : 10 correspondant au nombre de sommets, 3 au nombre de coordonnées (x, y, z), 9 au nombre de faces, 4 au nombre maximal de sommets pour une face.

La figure 1 vous montre le polyèdre choisi qui représente une maison dans le repère $Oxyz$. Les numéros des sommets sont encadrés.

En ligne 50 sont stockées les coordonnées des sommets dans l'ordre des numéros des sommets. Les groupes de 3 nombres sont séparés par une barre afin de rendre la lecture plus facile. Ces coordonnées sont introduites dans un tableau en ligne 80 . p (k, n) est la n ième coordonnée du sommet numéro k.

Les lignes 100 et 110 recadrent le polyèdre. Les valeurs 100 et 60 peuvent être modifiées suivant les besoins.

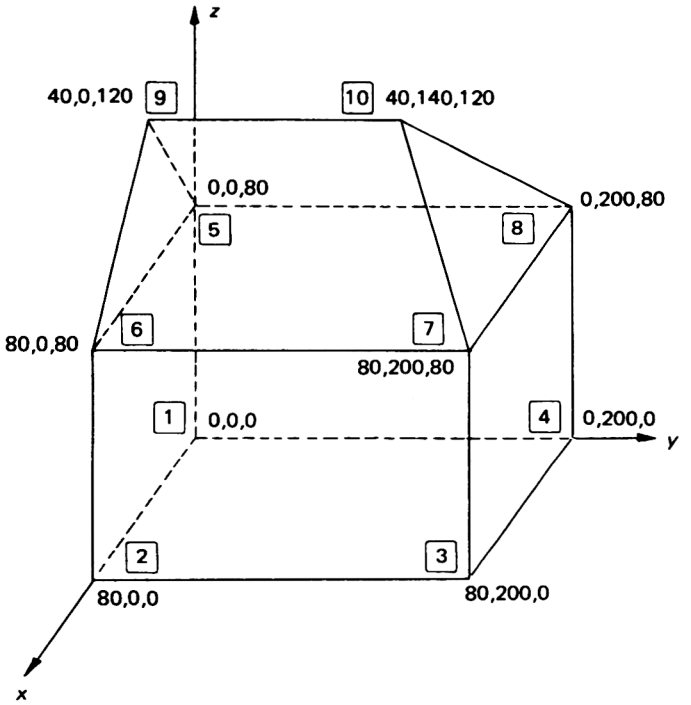


Fig. 1.

En ligne 140 sont stockées les définitions des faces. Par exemple, la troisième face est limitée par les sommets 2-3 7 6, l'ordre de ces nombres correspond à une rotation autour de la face. Si une vis était vissée dans cette face et si on la tournait dans ce sens, elle sortirait de la face vers l'extérieur du polyèdre (figure 2). Il en est de même pour les autres faces. Dans le cas des faces triangulaires deux sommets, les deux derniers du groupe de 4, sont confondus. En ligne 170, ces données sont introduites dans un tableau : *f* (*kk,n*) est le *n* ième sommet de la face *kk.*

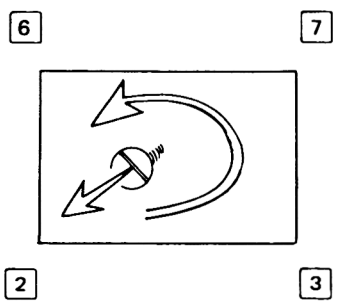


Fig. 2.

Perspective

En ligne 220 à 240 sont introduits les angles de vue, on calcule ensuite leurs lignes trigonométriques. En 270 sont calculés les coefficients directeurs (non normalisés) de la direction R du regard (figure 3).

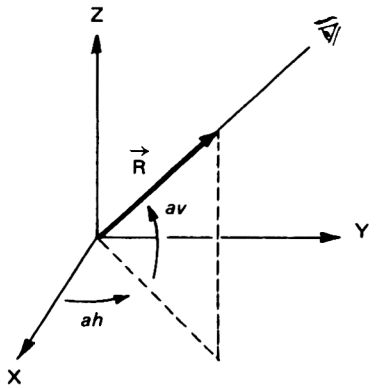


Fig. 3.

En ligne 240 est demandée l'échelle de reproduction que vous adapterez à vos besoins.

Regarder le polyèdre dans la direction R, c'est le placer dans un nouveau repère. Les coordonnées q (k,n) des sommets dans ce nouveau repère sont obtenues en ligne 300 à 320 par une matrice de rotation. Le travail est semblable à celui réalisé dans le programme «surface z (x, y)», nous n'y revenons donc pas.

Arêtes cachées

Soient A_1, A_2, A_3, A_4 les sommets d'une face. Nous allons maintenant calculer les coefficients directeurs de la normale à cette face. Cette normale est de même direction et de même sens que le produit vectoriel $N = A_1A_2 \wedge A_1A_4$ (figure 4). Les composantes des vecteurs A_1A_2 et A_1A_4 sont calculées en lignes 370 à 420 et les composantes du produit vectoriel en lignes 430 à 450.

Calculons enfin le produit scalaire T de ce vecteur N et du vecteur R (dans la direction du regard). En observant le signe de ce produit scalaire nous pouvons savoir si l'angle entre ces deux vecteurs est aigu ou obtus. S'il est aigu, nous voyons la face et nous la dessinons (lignes 490 à 510). S'il est obtus, c'est que la face est de l'autre côté du polyèdre, elle est cachée, nous ne la dessinons pas (ligne 480). Remarquons que dans ces calculs le module des vecteurs n'intervient pas, seuls interviennent la direction et le sens.

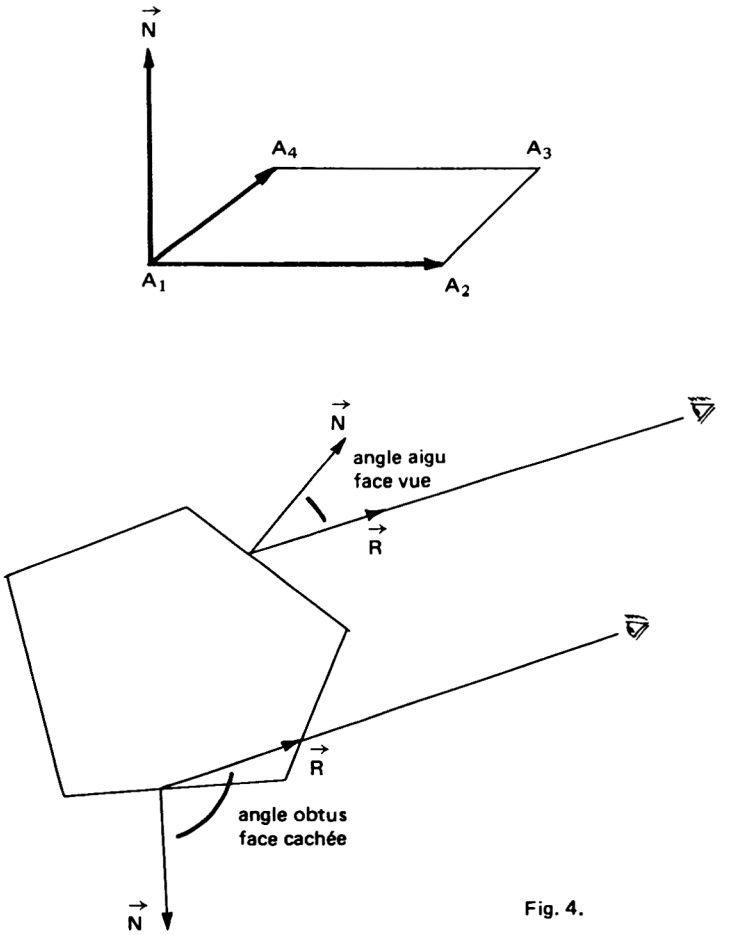


Fig. 4.

Illustrations

Souvenez-vous enfin que l'imprimante déforme les figures (voir «graphes de fonction, ... en polaires).

Les illustrations obtenues correspondent à :

Figure 5 :	*ah*=0	*av*=0	*e*=2
6 :	90	0	2
7	0	90	2
8	45	0	2
9	−45	−30	2
10	30	30	1,8

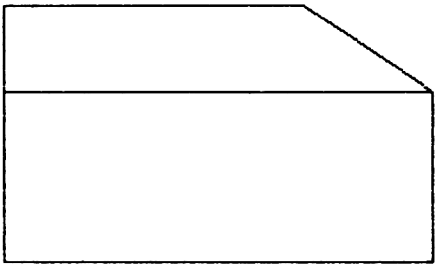


Fig. 5.

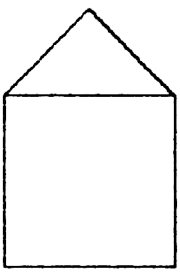


Fig. 6.

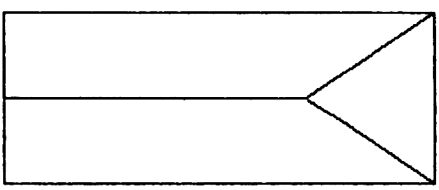


Fig. 7.

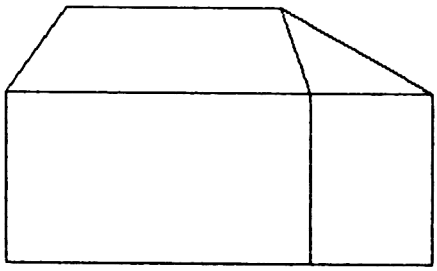


Fig. 8.

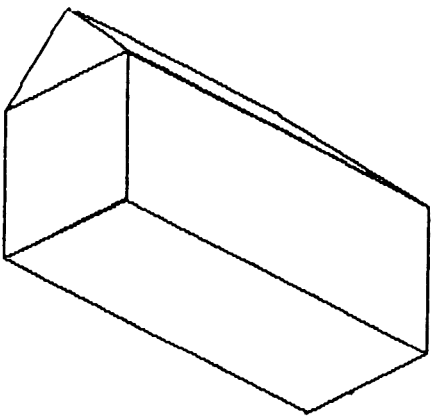


Fig. 9.

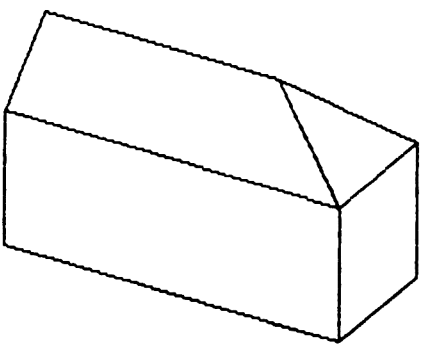


Fig. 10.

Programme

```
10 REM"polyedre"
20 REM
30 CLS
40 DIM p(10,3):DIM q(10,3):DIM f(9,4)
50 DATA 0,0,0,/,80,0,0,/,80,200,0,/,0,200,0,/,0,0,
80,/,80,0,80,/,80,200,80,/,0,200,80,/,40,0,120,/,4
0,140,120,/
60 REM
70 FOR k=1 TO 10
80 READ p(k,1),p(k,2),p(k,3),x$
90 REM
100 p(k,2)=p(k,2)-100
110 p(k,3)=p(k,3)-60
120 NEXT k
130 REM
140 DATA 1,4,3,2,/,1,2,6,5,/,2,3,7,6,/,3,4,8,7,/,4
,1,5,8,/,5,6,9,9,/,7,8,10,10,/,9,6,7,10,/,10,8,5,9
,/
150 REM
160 FOR kk=1 TO 9
170 READ f(kk,1),f(kk,2),f(kk,3),f(kk,4),x$
180 f(kk,0)=f(kk,4)
190 NEXT kk
200 REM
210 DEG
220 INPUT "ah=";ah
230 INPUT "av=";av
240 INPUT "e=";e
250 sh=SIN(ah):ch=COS(ah)
260 sv=SIN(av):cv=COS(av)
270 rx=cv*ch:ry=cv*sh:rz=sv
280 REM
290 FOR k=1 TO 10
300 x=p(k,1)*ch+p(k,2)*sh
310 q(k,1)=-p(k,1)*sh+p(k,2)*ch
320 q(k,2)=-x*sv+p(k,3)*cv
330 NEXT k
340 REM
350 CLS
360 FOR kk=1 TO 9
370 x2=p(f(kk,2),1)-p(f(kk,1),1)
380 x4=p(f(kk,4),1)-p(f(kk,1),1)
390 y2=p(f(kk,2),2)-p(f(kk,1),2)
400 y4=p(f(kk,4),2)-p(f(kk,1),2)
410 z2=p(f(kk,2),3)-p(f(kk,1),3)
420 z4=p(f(kk,4),3)-p(f(kk,1),3)
430 nx=y2*z4-z2*y4
440 ny=z2*x4-x2*z4
450 nz=x2*y4-y2*x4
460 t=rx*nx+ry*ny+rz*nz
470 REM
480 IF t<0 THEN GOTO 530
490 FOR n=0 TO 3
500 MOVE 320+e*q(f(kk,n),1),200+e*q(f(kk,n),2)
510 DRAWR e*(q(f(kk,n+1),1)-q(f(kk,n),1)),e*(q(f(k
k,n+1),2)-q(f(kk,n),2))
520 NEXT n
530 NEXT kk
540 END
```

Fractal

Ce mot recouvre une technique permettant de dessiner des courbes et des surfaces presque émouvantes car elles imitent des reliefs naturels de façon étonnamment réaliste.

Ces courbes ont un point commun avec les horizons ou les rivages, c'est que lorsqu'on les détaille (lorsqu'on augmente le nombre de points calculés sur une courbe allant d'un point donné à un autre point donné) la longueur de la courbe augmente et peut tendre vers l'infini. A la limite ces courbes peuvent remplir des surfaces (flocon de Von Koch, courbe de Péano). Le mathématicien Mandelbrot leur a attribué une dimension supérieure à 1 (une «fraction» d'où «fractal»).

Remarquons tout d'abord que la fonction RND fournit à chaque mise sous tension de l'Amstrad toujours la même suite de valeurs. Vous pouvez au besoin la faire tourner par :

```
FOR X = TO ... : Y = RND : NEXT X
```

Courbe fractale

Nous allons déterminer 257 points de cette courbe de $x = 0$ à $x = 256$. Le cadrage est prévu pour y variant de 0 à 400. Il se peut toutefois que la courbe sorte de l'écran. Les valeurs 0, 200 en ligne 40 et 99 en ligne 80 ne sont que des conseils.

On demande d'abord de fixer les extrémités et le milieu de la courbe (ligne 110), ensuite on calcule le milieu de chacun de ces deux segments en ligne 150. Mais on s'écarte de ces points d'une quantité prise au hasard entre $-e$ et $+e$ (fig. 1). On obtient ainsi deux nouveaux points de la courbe. On reprend le même processus sur les quatre segments obtenus, mais la quantité prise au hasard est comprise cette fois-ci entre $-e/2$ et $+e/2$, ainsi de suite.

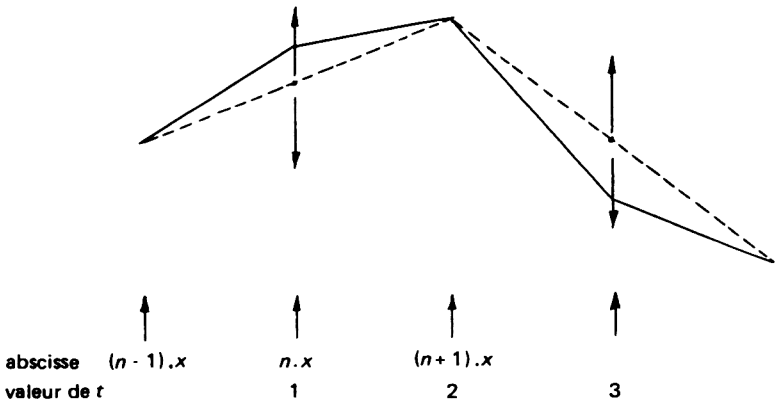


Fig. 1

Plus le segment à partager est petit, plus la quantité prise au hasard est petite. On peut parler d'échelle pour le hasard. La courbe finale s'écarte d'autant plus de la courbe initiale que *e* est grand.

S est le numéro de la boucle de calcul
X est l'écart entre deux points
T numérote les points

Les exemples suivants ont été pris avec :

Figure 2 :	$a=100$	$b=100$	$c=100$	$e=20$
3	100	100	100	90
4	0	200	0	99

Si la figure 2 évoque un horizon du massif central, la figure 5 évoque plutôt les Alpes...

Fig. 2.

Fig. 3.

Fig. 4.

Premier programme

```
10 REM"courbe fractale"
20 REM
30 CLS
40 PRINT:PRINT"nombres compris entre 0 et 200"
50 INPUT"a=";a
60 INPUT"b=";b
70 INPUT"c=";c
80 PRINT:INPUT"degre (entre 0 et 99)";e
90 CLS:PRINT"calcul en cours"
100 DIM y(256)
110 y(0)=a:y(128)=b:y(256)=c
120 t=3:x=64
130 FOR s=1 TO 7
140 FOR n=1 TO t STEP 2
150 y(n*x)=(y((n-1)*x)+y((n+1)*x))/2
160 y(n*x)=y(n*x)+e*(-1+2*RND)
170 NEXT n
180 t=2*t+1:x=x/2:e=e/2
190 NEXT s
200 CLS
210 FOR x=1 TO 256
220 PLOT 80+2*x,y(x)+100
230 DRAWR -2,y(x-1)-y(x)
240 NEXT x
250 END
```

Surface fractale

Le «%» est destiné à alléger la mémoire de l'ordinateur.

On représente la surface y (u, v), u et v étant compris entre 0 et 64. 64 fois 64 valeurs de y sont donc à calculer. On demande d'abord à l'utilisateur de choisir les valeurs de y pour les sommets et le centre du carré u, v. Les autres valeurs sont calculées ensuite de la même façon

que pour la courbe. On détermine d'abord le milieu des droites côtés de chaque maille carrée puis leur centre comme le montre la figure apparaissant sur l'écran.

La surface obtenue est représentée en perspective suivant la technique évoquée dans le programme « représentation d'une surface z (x, y) ». Nous n'y revenons donc pas ici.

Nous avons fixé *ah* à 0. Nous n'avons pas envisagé de représenter le dessous de la surface mais vous savez comment le faire.

Dans l'idée d'évoquer des paysages, nous avons choisi d'annuler toutes les valeurs de y négatives, ceci est fait en lignes 415 à 417, nous obtenons ainsi des images de rochers au bord de la mer.

Les figures suivantes ont été obtenues avec :

Figure 5 :	$a=-50$	$b=-50$	$c=200$	$d=-50$	$e=-50$	$ee=80$	$av=20$
6	−50	−50	300	−50	−50	200	20
7	0	−100	0	200	−100	200	20

Organisation du programme

Ligne(s)	
10	introduction des données
40 à 370	calcul des valeurs de y
210	détermination des milieux des côtés
310	détermination des centres des carrés
390	tracé de la surface en perspective

Remarque : dans ce deuxième programme, la valeur maximale de la quantité prise au hasard est notée *ee*.

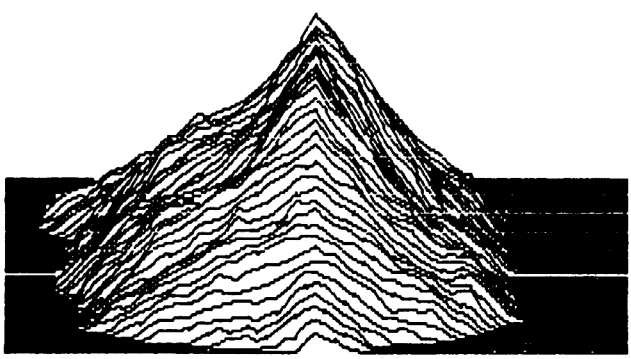


Fig. 5.

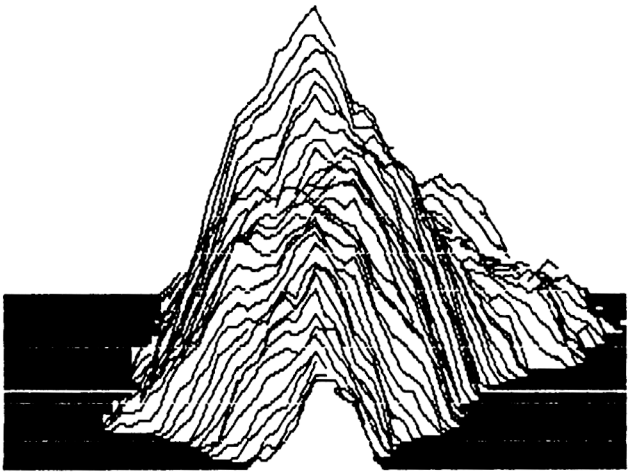


Fig. 6.

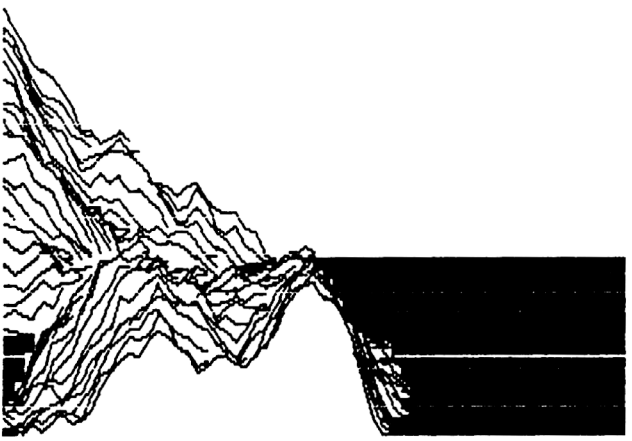


Fig. 7.

Deuxième programme

```
10 REM"surface fractale"
20 REM
30 CLS
50 PRINT:PRINT"d      e":PRINT:PRINT"    c":PRINT:PR
INT"a      b"
60 PRINT:INPUT"a=";a
70 INPUT"b=";b
80 INPUT"c=";c
90 INPUT"d=";d
100 INPUT"e=";e
110 PRINT:INPUT"degre ";ee
```

```
120 DEG: PRINT: INPUT "angle (0< <90)";av
130 CLS
140 DIM y%(64,64)
150 y%(0,0)=a:y%(64,0)=b
160 y%(0,64)=d:y%(64,64)=e
170 t=1:x=32:PLOT 192,192
180 FOR s=1 TO 6
190 FOR v=0 TO 64 STEP 2*x
200 FOR n=1 TO t STEP 2
210 y%(n*x,v)=((y%((n-1)*x,v)+y%((n+1)*x,v))/2 )+e
e*(-1+2*RND)
220 y%(v,n*x)=((y%(v,(n-1)*x)+y%(v,(n+1)*x))/2 )+e
e*(-1+2*RND)
250 PLOT 6*n*x,6*v:PLOT 6*v,6*n*x
260 NEXT n
270 NEXT v
280 FOR n=1 TO t STEP 2
290 FOR nn=1 TO t STEP 2
300 y%(n*x,nn*x)=(0.25*(y%(n*x+x,nn*x)+y%(n*x-x,nn
*x)+y%(n*x,nn*x+x)+y%(n*x,nn*x-x)))+ee*(-1+2*RND)

320 PLOT 6*n*x,6*nn*x
330 NEXT nn
340 NEXT n
350 y%(32,32)=c
360 t=2*t+1:x=x/2:ee=ee/2
370 NEXT s
380 REM
390 CLS:PRINT"calcul en cours"
410 sh=SIN(av):ch=COS(av)
415 FOR w=0 TO 64:FOR z=0 TO 64
416 IF y%(w,z)<0 THEN y%(w,z)=0
417 NEXT z:NEXT w
420 CLS
430 DIM m(65)
440 FOR w=0 TO 64
450 k=0
460 FOR z=0 TO 64
470 xe=6*z+128
480 ye=6*sh*w+y%(z,w)*ch
490 IF ye<m(z) THEN GOTO 540
500 PLOT xe,ye
510 m(z)=ye
520 IF k=1 THEN DRAWR -6,yy-ye
530 k=1
540 yy=ye
550 NEXT z
560 NEXT w
```

Méthode des moindres carrés
Applications aux équations non linéaires

En physique, les graphiques expérimentaux présentent en général des irrégularités. Si les points représentés se répartissent suivant une configuration assez régulière, le physicien ajustera la ou les grandeurs étudiées ; c'est-à-dire qu'il substituera aux grandeurs observées des grandeurs calculées.

La première étape de l'ajustement porte sur le choix de la courbe d'ajustement, la seconde étape porte sur la détermination de l'équation de la courbe d'ajustement. Nous nous intéresserons aux équations de la forme : $Y = f(X)$.

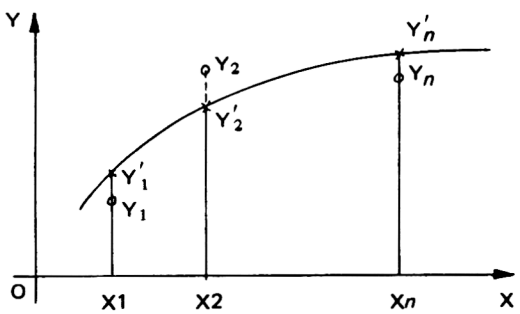
Méthode des moindres carrés

C'est une méthode analytique qui permet de calculer les différents paramètres W_i d'une équation de la forme $Y = f(X)$.

Représentons graphiquement les points expérimentaux de coordonnées :

$(X1, Y1)$ $(X2, Y2)$ ..... (Xn, Yn)

et les points de coordonnées qui se trouvent sur la courbe d'ajustement, soit $(X1, Y'1)$ $(X2, Y'2)$ ..... $(Xn, Y'n)$



La fonction $Y = f(X)$ dépendant de plusieurs paramètres W_i des valeurs différentes de ces paramètres conduisent à des fonctions $Y = f(X)$ différentes.

La méthode des moindres carrés consiste à retenir la fonction $Y = f(X)$ qui rend la plus faible possible la somme des carrés des écarts : $Y_i - Y'_i$.

Il est hors de question pour nous de donner, dans cet ouvrage, la démonstration mathématique rigoureuse du calcul des différents paramètres W_i de la fonction $Y = f(X)$.

Nous utiliserons la méthode de DEMING (1), reprise par WENTWORTH (2), le lecteur intéressé pourra éventuellement consulter les articles des auteurs.

Supposons que la fonction d'ajustement contienne la variable X et un certain nombre de paramètres : W1, W2 ... WJ : soit Y' (X, W1 ... WJ).

Il s'agit de trouver à l'aide d'un calcul itératif les valeurs optimisées des différents paramètres W_i (somme des carrés des écarts $Y_i - Y'_i$ minimale).

Soit ΔW_i la différence observée entre deux itérations successives pour le paramètre W_i.

Ecrivons que : $\Delta W_i = \mathring{W}_i - W_i$ et $Y'_i = Y(X_i, \mathring{W1}, \mathring{W2} \ldots \mathring{WJ})$.

Posons $F(W_i) = \left[\dfrac{\partial (Y_i - Y'_i)}{\partial W_i}\right] X_i, \mathring{W1}, \mathring{W2} \ldots \mathring{WJ}$

DEMING (1) a démontré que si les X_i sont connus avec précision, les ΔW_i se calculent en résolvant le système d'équations suivant.

En posant $\mathring{F}_i = Y_i - Y'_i$:

$$\sum_1^J F(W1).F(W1).\ W1 + \sum_1^J F(W1).F(W2).\ W2 + \ldots + \sum_1^J F(W1.F(WJ).\ WJ = \sum_1^J F(W1).\mathring{F}_i$$

$$\sum_1^J F(W2).F(W1).\ W1 + \sum_1^J F(W2).F(W2).\ W2 + \ldots + \sum_1^J F(W2).F(WJ).\ WJ = \sum_1^J F(W2).\mathring{F}_i$$

$$\vdots$$

$$\sum_1^J F(WJ).F(W1).\ W1 + \sum_1^J F(WJ).F(W2).\ W2 + \ldots + \sum_1^J F(WJ).F(WJ).\ WJ = \sum_1^J F(WJ).\mathring{F}_i$$

Liste des variables

N	nombre de points expérimentaux
J	nombre de coefficients de l'équation d'ajustement
X (N)	tableau pour les différents X*i*
Y (N)	" " " " Y*i*
V (K)	tableau pour les dérivés F (W*i*) **Remarque :** V (1) = Y (I) − Y'
Z (J, J)	tableau pour les différents éléments de la matrice principale
COR	coefficient de corrélation

Déroulement du programme

Ligne(s)	
1 à 20	entrée des données (N et J)
40 à 130	entrée de chaque couple de points expérimentaux (X*i*, Y*i*) **Remarque :** à la ligne 70, on rentre les différentes valeurs des coefficients W*i* de départ (choix de l'utilisateur)
140 et 150	on repère les valeurs maximales en valeur absolue des X*i* et des Y*i*
200 à 250	initialisation des différents éléments de la matrice
340	on pose V (1) = Y (I) − Y'
370	on pose DEF FNZ (W) = Y (I) − Y'
390 à 410	les différentes dérivées F (W*i*) = V (L + 1) sont calculées numériquement
341	on calcule le coefficient de corrélation COR après chaque itération
500	saut au sous-programme d'inversion de matrice

Les calculs se déroulent sous vos yeux jusqu'à la convergence du coefficient de corrélation (noter les différents W*i* après avoir arrêté le programme en appuyant 2 fois sur ESC à la convergence).

1000 à 1200	l'ordinateur efface l'écran. Il demande à nouveau les différents W*i* calculés à la convergence
1025	on trace les axes, les points expérimentaux à l'échelle et la courbe d'ajustement **Remarque :** en ligne 1080, n'oubliez pas de rentrer la fonction d'ajustement Y'

```
1 CLS
5 REM MOINDRES CARRES
6 REM ***************
7 REM
10 INPUT "NB DE POINTS EXP.N=";N
20 INPUT "NB DE COEFFTS.J=";J
35 DIM X(N):DIM Y(N):DIM Z(J,J):DIM D(J)
40 M=0:P=0:DIM W(J+1):DIM R(J)
42 K=J+1:DIM V(K)
44 DIM U(J,J):DIM E(J):DIM F(J)
50 DW=0.001:PR=0.00000001:MI=1E-30
55 MODE 2
60 FOR I=1 TO J
70 INPUT "W=";W(I):PRINT "W";I;"=";W(I)
80 NEXT I
90 CLS
100 FOR I=1 TO N
110 PRINT "I=";I
120 INPUT "X=";X(I):INPUT "Y=";Y(I)
130 PRINT "X";I;"=";X(I);":";"Y";I;"=";Y(I):PRINT
140 IF ABS(Y(I))>P THEN P=ABS(Y(I))
150 IF ABS(X(I))>M THEN M=ABS(X(I))
160 NEXT I
190 S=0
200 FOR C=1 TO J
210 FOR L=1 TO J
220 Z(L,C)=0
230 NEXT L
240 NEXT C
250 FOR L=1 TO J:D(L)=0:NEXT L
300 REM CALCUL ITERATIF
310 REM ***************
316 FOR I=1 TO N
320 X=X(I)
340 V(1)=Y(I)-(W(1)*(1-X)-W(1)*(1-X)*EXP(-W(2)*X/(
1-X))+W(3)*X-W(3)*X*EXP(-W(4)*(1-X)/X))
341 D=V(1)*V(1):S=S+D
350 FOR L=1 TO J
360 W=W(L):A=W(L)
370 DEF FNZ(W)=Y(I)-(W(1)*(1-X)-W(1)*(1-X)*EXP(-W(
2)*X/(1-X))+W(3)*X-W(3)*X*EXP(-W(4)*(1-X)/X))
390 W(L)=A+DW:Z1=FN Z(W)
400 W(L)=A-DW:Z2=FN Z(W)
410 V(L+1)=(Z1-Z2)/(2*DW)
411 B=V(L+1)*V(1)
412 D(L)=D(L)+B
415 W(L)=A
420 NEXT L
440 FOR C=1 TO J
445 FOR L=1 TO J
450 H=V(L+1)*V(C+1)
455 Z(L,C)=Z(L,C)+H
460 NEXT L
470 NEXT C
490 NEXT I
491 CR=S/N
492 PRINT "COR=";SQR(CR)
493 PRINT
500 GOSUB 2021
600 FOR L=1 TO J:R(L)=0
```

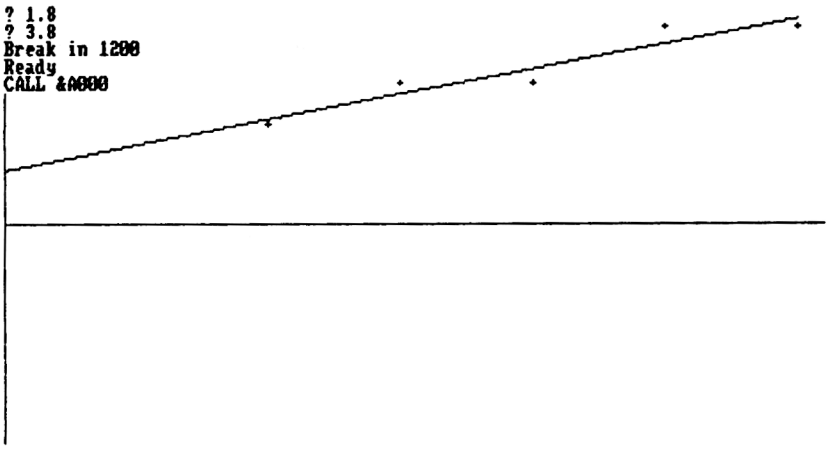
```
610 FOR C=1 TO J
620 R(L)=R(L)+Z(L,C)*D(C)
630 NEXT C
640 NEXT L
641 FOR F=1 TO J
642 W(F)=W(F)-R(F):PRINT W(F)
643 NEXT F
660 GOTO 190
1000 CLS
1010 PLOT 0,200:DRAWR 620,0
1020 PLOT 0,0:DRAWR 0,400
1025 FOR F=1 TO J:INPUT W(F):NEXT F
1030 FOR I=1 TO N
1040 PLOT X(I)/M*600,Y(I)/P*180+200:DRAWR 2,0
1041 PLOT X(I)/M*600,Y(I)/P*180+200:DRAWR 0,2
1042 PLOT X(I)/M*600,Y(I)/P*180+200:DRAWR -2,0
1043 PLOT X(I)/M*600,Y(I)/P*180+200:DRAWR 0,-2
1050 NEXT I
1060 FOR K=1 TO 600
1070 X=(M/600)*K
1080 Y=W(1)*(1-X)-W(1)*(1-X)*EXP(-W(2)*X/(1-X))+W(
3)*X-W(3)*X*EXP(-W(4)*(1-X)/X)
1085 IF ABS(Y/P*180)>200 THEN GOTO 1100
1090 PLOT K,Y/P*180+200
1100 NEXT K
1200 STOP
2000 REM SP.INVERSION DE MATRICE
2010 REM **********************
2021 FOR C=1 TO J
2022 FOR L=1 TO J
2023 U(L,C)=Z(L,C)
2025 NEXT L
2026 NEXT C
2030 DET=1
2031 FOR G=1 TO J
2035 MA=MI
2036 FOR L=G TO J
2037 FOR C=G TO J
2040 A=Z(L,C)
2045 IF ABS(A)<=MA THEN GOTO 2050
2046 IM=L:JM=C:MA=ABS(A)
2050 NEXT C
2060 NEXT L
2070 IF MA<>MI THEN GOTO 2100
2080 STOP
2100 PT=Z(IM,JM):DET=DET*PT
2110 F(G)=IM:E(G)=JM
2120 IF IM=G THEN GOTO 2160
2130 FOR C=1 TO J
2140 A=Z(IM,C):Z(IM,C)=Z(G,C):Z(G,C)=A
2150 NEXT C
2160 IF JM=G THEN GOTO 2200
2170 FOR L=1 TO J
2180 A=Z(L,G):Z(L,G)=Z(L,JM):Z(L,JM)=A
2190 NEXT L
2200 FOR L=1 TO J
2210 IF L=G THEN GOTO 2300
2220 R=Z(L,G)/PT
2230 FOR C=1 TO J
2240 IF C=G THEN GOTO 2290
```

```
2250 DD=Z(L,C)-R*Z(G,C)
2260 IF ABS(DD)<ABS(Z(L,C))*PR THEN DD=0
2270 Z(L,C)=DD
2290 NEXT C
2300 NEXT L
2400 FOR Z=1 TO J
2410 Z(Z,G)=Z(Z,G)/PT
2420 Z(G,Z)=-Z(G,Z)/PT
2430 NEXT Z
2500 Z(G,G)=1/PT
2510 NEXT G
2520 FOR G=1 TO J
2530 Z=J-G+1
2540 PL=F(Z)
2550 IF PL=Z THEN GOTO 2600
2560 FOR L=1 TO J
2570 A=Z(L,Z):Z(L,Z)=Z(L,PL):Z(L,PL)=A
2580 NEXT L
2600 PC=E(Z)
2610 IF PC=Z THEN GOTO 3000
2620 FOR C=1 TO J
2630 A=Z(Z,C):Z(Z,C)=Z(PC,C):Z(PC,C)=A
2640 NEXT C
3000 NEXT G
3010 RETURN
```

Nous donnons 3 exemples d'utilisation de ce programme.

Exemple 1 : la fonction d'ajustement est une droite.

```
 1.79999942
 3.8000015
COR= 0.848528138

 1.8
 3.8
COR= 0.848528138

 1.8
 3.8
COR= 0.848528138

 1.8
 3.8
COR= 0.848528138

 1.8
 3.8
COR= 0.848528138

 1.8
 3.8
Break in 341
Ready
CALL &A000
```



$$Y' = W(1) * X + W(2)$$

Rentrer au départ W (1) = 1 et W (2) = 1

X	Y
2	7
3	10
4	10
5	14
6	14

N = 5

J = 2

Exemple 2 : la fonction d'ajustement est une fonction du cinquième degré rendant compte de l'enthalpie de mélange pour le propanol-eau.

Mélange 1 PROPANOL
Mélange 2 EAU

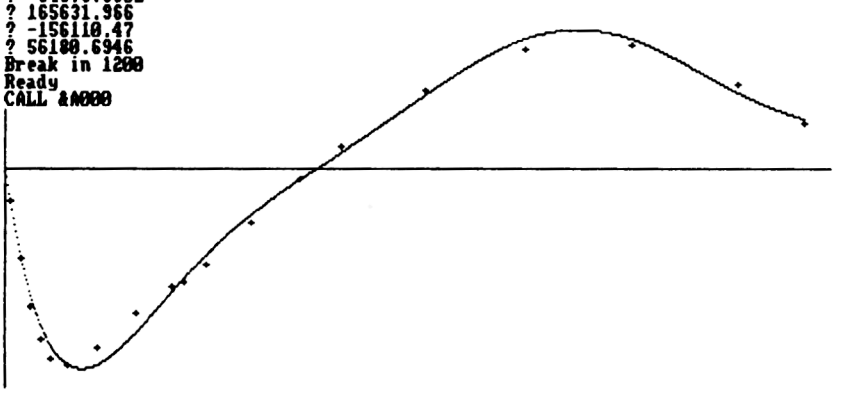
x	*Hm*
0.0059	− 12.9
0.0177	− 36.9
0.0291	− 56.5
0.0403	− 70.5
0.051	− 78.5
0.071	− 81.3
0.105	− 73.4
0.149	− 59.5
0.189	− 48.8
0.203	− 46.6
0.229	− 39.8
0.280	− 22.6
0.337	− 4.1
0.384	+ 9.6
0.478	+ 32.6
0.592	+ 49.7
0.714	+ 51.9
0.833	+ 35.7
0.909	+ 18.9

LEYDET (3) a montré que les fonctions de mélange en thermodynamique pouvaient être représentées par des fonctions d'exponentielle. Essayez les mêmes valeurs itérées avec la fonction donnée en lignes 340-370 et 1080 dans le listing.

Remarque : les fonctions de mélange s'annulent pour $x = 0$ et $x = 1$.

```
-2431.11528
 22232.7981
-84975.8302
 165631.966
-156110.47
 56180.6946
COR= 3.9747091

-2460.86178
 22842.0333
-88583.921
 174676.533
-165572.972
 59713.4817
COR= 6.02328216

-2483.9391
 22953.5447
-88062.0372
 171447.537
-161712.434
 58588.1611
Break in 400
Ready
CALL &A000
```

```
? -2431.11528
? 22232.7981
? -84975.8302
? 165631.966
? -156110.47
? 56180.6946
Break in 1200
Ready
CALL &A000
```



$$Y' = X(1 - X)(W(1) + W(2) * X + W(3) * X \uparrow 2 + W(4) * X \uparrow 3 + W(5) * X \uparrow 4 + W(6) * X \uparrow 5)$$

On part de $W(1) = W(2) = \ldots = W(6) = 1$

Essayer ensuite la fonction : (donnée dans le listing)

$$Y' = W(1) * (1 - X) - W(1) * (1 - X) * \mathrm{Exp}(-W(2) * X / (1 - X)) + W(3) * X - W(3) * X * \mathrm{Exp}(-W(4) * (1 - X) / X)$$

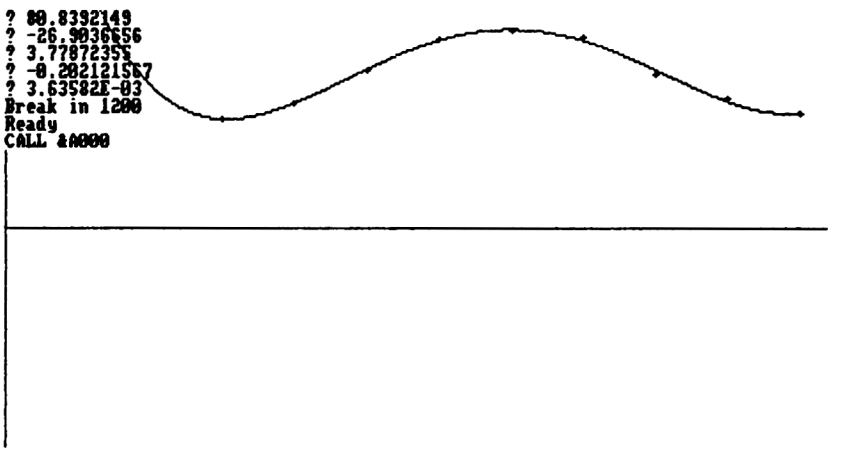
partir de : $W(1) = -100$ $W(2) = 30$
$W(3) = 300$ et $W(4) = 3$

Exemple 3 : lissage d'une courbe de températures relevées à Marseille au mois de juillet.

```
 80.8757324
-26.9172363
 3.78041077
-0.202210903
 3.63743E-03
COR= 0.343686623

 80.8628021
-26.9117399
 3.77968136
-0.202168843
 3.63665E-03
COR= 0.343493226

 80.8392149
-26.9036656
 3.77872355
-0.202121567
 3.63582E-03
COR= 0.343493402

Break in 2260
Ready
CALL &A000
```

x (heures)	y (°C)
6	16,5
8	18,9
10	23,8
12	28,4
14	29,6
16	28,7
18	23,1
20	19,4
22	17,3

```
? 80.8392149
? -26.9036656
? 3.77872355
? -0.202121567
? 3.63582E-03
Break in 1200
Ready
CALL &A000
```



```
Y' = W(1) + W(2) * X + W(3) * X↑2 + W(4) * X↑3 + W(5) * X↑4
```

BIBLIOGRAPHIE

(1) **DEMING, W.E.,** *« Statistical Adjustment of Data »*, John Wiley and Sons, Inc., New-York, 1943.

(2) **WENTWORTH, W.E.,** *« Journal of Chemical Education »*, Volume 4,2, Number 2, February 1965.

(3) **LEYDET, P.,** *« Thermochimica Acta »*, 55 (1982) 193-200.

Composition-Maquette
GRAPHI-DACTYL
27400 LOUVIERS

Imprimé en France. — JOUVE, 18, rue Saint-Denis, 75001 PARIS
N° 16131. Dépôt légal : Septembre 1986
N° d'éditeur : 4542

EYROLLES

PROGRAMMES DE MATHEMATIQUES SUR AMSTRAD

P. BEAUFILS
M. LAMARCHE
Y. MUGGIANU

Voici des programmes de mathématiques écrits en BASIC.

Accompagnés de leurs commentaires, ils transforment l'ordinateur en un outil pédagogique de valeur et un instrument de calcul puissant pour résoudre bien des problèmes.

En second plan, les auteurs ont voulu montrer que les mathématiques ne sont pas mortes avec l'usage de l'ordinateur, bien au contraire : l'écriture d'un programme comme la critique des résultats nécessite des connaissances mathématiques solides et variées.

Cet ouvrage original devrait intéresser et surtout rendre service au lecteur, pourvu qu'il ait le niveau mathématique d'une classe terminale scientifique.

couverture JÉRÔME LO MONACO

EYROLLES

8346

Document numérisé
avec amour par :
AMSTRAD
CPC
MÉMOIRE ÉCRITE
ACME
https://acpc.me/